Universidad Zaragoza
1542

Trabajo Fin de Grado

Sincronización de textos asociados a archivos audiovisuales

Synchronization of texts associated with audiovisual files

Autor

Laura Pampliega Sanz

Director

Eduardo Lleida Solano

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2020

AGRADECIMIENTOS

Principalmente a Eduardo Lleida por su trabajo, ayuda y atención en absolutamente todo momento.

A mi familia por su gran apoyo en estos años de carrera y a los amigos que he conocido en la carrera, por el apoyo mutuo y con los que he pasado grandes momentos.

Sincronización de textos asociados a archivos audiovisuales

RESUMEN

El tema principal es la sincronización de subtítulos. Los subtítulos deben estar formados por el mismo contenido y orden del texto original, que se corresponden con las transcripciones del archivo audiovisual que se desea sincronizar. El fin de este TFG es la sincronización de los subtítulos de la manera más automática posible. La sincronización se realiza utilizando un sistema de reconocimiento de voz controlado por un modelo de lenguaje (gramática) construido a partir de los textos a sincronizar.

En este trabajo se evalúa el uso de dos tipos de gramáticas en un primer intento de reconocimiento, estocástica y basada en estados finitos, que permiten cierta libertad en la formación de frases y como consecuencia, permiten al reconocedor la sincronización texto-voz de una manera parcialmente libre. A continuación, se realiza una alineación del texto referencia con la hipótesis aportada por el reconocedor. Esta alineación de palabras permite determinar los fragmentos de texto y audio erróneamente detectados. Obtenidos unos puntos de anclaje, zonas correctamente detectadas, se trata de forzar estas zonas conflictivas a través de una gramática forzada. Se analizan las distintas gramáticas y se decide el método que ofrece mejores resultados. Se ha analizado la relación de los resultados con las calidades acústicas de los distintos programas a disposición, lo que se traduce en una mayor o menor dificultad de sincronización. Por último, se muestran las conclusiones de este TFG.

Synchronization of texts associated with audiovisual files

ABSTRACT

The topic is the synchronization of subtitles. The subtitles must be made up of the same content and order of the original text, which corresponds to the transcripts of the audiovisual file which is wanted to be synchronized. The purpose of this TFG is the synchronization of the subtitles in the most automatic way possible. The synchronization is made using a voice recognition system controlled by a language model (grammar) built with the transcripts.

This work evaluates the use of two types of grammar in a first attempt at recognition, stochastic and based on finite states, which allows some freedom in the formation of sentences and as a consequence, allows the recognizer to do the text-speech synchronization in a partially free way. Then, an alignment of the reference text is made with the hypothesis provided by the recognizer. This word alignment allows to determine the erroneously detected fragments of text and audio. Obtained the anchor points (correctly detected areas), these conflictive areas with a forced grammar are forced. The different grammars are analysed and the method that offers the best results is decided. The relation of the results with the acoustic qualities of the different programs available has been analysed too, which translates into more or less difficulty. Finally, the conclusions of this TFG are shown.

Índice

CAPÍTULO 1. Introducción

1.1. Motivación

El habla es la forma de comunicación más utilizada por el ser humano. La tecnología basada en el tratamiento o reconocimiento de voz es compleja, puesto que la voz tiene muchos matices tales como la entonación, el sexo del hablante, la velocidad a la que se habla u otras variantes a nivel acústico. Además, las reglas que sigue el lenguaje pueden variar de manera drástica en función del idioma o la región en la que vive el hablante. En los últimos años, los avances en procesamiento del habla han sido muy importantes y con notables resultados, por lo que es un campo de trabajo muy complejo aunque también atrayente en el que desarrollar y experimentar.

Una aplicación llamativa es la realización de búsquedas en audios o vídeos con el fin de encontrar de una manera más sencilla un intervalo de tiempo de interés para el usuario. De manera más concreta, un ejemplo serían los audiolibros, donde una búsqueda rápida en el audio podría ser de gran utilidad. Otras alternativas podrían ser el control de dispositivos por voz como teléfonos móviles o asistentes virtuales, actualmente en auge. Con lo que respecta a la sincronización, tema clave en ese TFG, también existen estudios muy interesantes a tratar en este campo. Una temática atrayente podría ser la resincronización, es decir, programas de televisión o radio en directo, donde un locutor reproduce las conversaciones que se llevan a cabo en el programa y genera los subtítulos en el mismo momento de la emisión. Estos subtítulos pueden llevar unos segundos de retraso debido a que este locutor debe escuchar y transmitir lo que se comenta en el programa. Si se realiza una segunda emisión de este programa podría ser interesante hacer una resincronización, donde los subtítulos se emitan en el momento preciso y no se encuentren desfasados. Otra alternativa se centra en el entrenamiento de los modelos, emplear los ficheros de subtítulos de diferentes tipos de audios para el entrenamiento de modelos acústicos. Estas son unas posibles aplicaciones, aunque al ser un campo de trabajo tan amplio y actual, existen inmensas posibilidades.

1.2. Objetivo

Para el desarrollo de este trabajo es necesario disponer de una base de datos en la que se disponga de archivos audiovisuales y sus correspondientes transcripciones. Las transcripciones deben estar compuestas de los textos en sí y de los hablantes que colaboran en el audiovisual. Partiendo desde este punto, el objetivo del trabajo es sincronizar las transcripciones asociando cada sección de texto al archivo audiovisual, es decir, agrupar cada segmento de texto, más el hablante y el intervalo de tiempo en el que se dice ese segmento en el audiovisual. El sistema desarrollado en este trabajo devuelve un fichero de subtítulos (extensión .srt) y un fichero de referencias de tiempo (extensión .stm). Empleando un programa que permita reproducir el vídeo o el audio del programa escogido y asociar a este unos subtítulos, los cuales serían este archivo con extensión .srt, se podrá observar de una manera más visual los textos sincronizados en el momento preciso en el que se narran en el audiovisual. La idea principal es obtener estos subtítulos, formato ampliamente utilizado, dado que existen múltiples programas y herramientas que permiten observar los resultados de una manera más visual, como Audacity en el caso de proporcionar el audio o el reproductor multimedia VLC en el caso de proporcionar el vídeo.

1.3. Herramientas utilizadas

Uno de los elementos principales es el reconocedor automático del habla. Se emplea el sistema de reconocimiento automático (RAH) del habla del grupo Vivolab. En el capítulo 2 “Reconocedor automático de habla” se explicará su funcionamiento de manera más exhaustiva, dado que es fundamental su uso para el desarrollo de este TFG.

Por otro lado, la programación de las herramientas necesarias se lleva a cabo con el lenguaje de programación Python.

Otra herramienta clave, es la base de datos de RTVE donde se obtienen los audios de los programas y sus transcripciones. Los programas con los que se trabaja se explican en el apartado 4.3 “Base de datos”.

Se experimenta con varias gramáticas. En la primera fase se necesita una herramienta para el entrenamiento de gramáticas estocásticas, por ello se utiliza la herramienta *SRILM* de libre distribución. También se integrará una herramienta de construcción de gramáticas de estados finitos y gramática forzada.

En la primera fase se aplica una gramática que permite cierta libertad en la formación de frases, por lo que se necesita una herramienta que sea capaz de determinar las zonas que han sido detectadas correctamente y las zonas erróneas. Para la alineación de secuencias de palabras, y poder diferenciar entre estas dos zonas, se utilizará el *toolkit SCTK (scoring toolkit) del NIST*.

1.4. Estado del arte

Se parte del programa *ZarCaptions* del grupo VivoLab. Este programa consiste en la generación de un fichero de subtítulos proporcionando las transcripciones y el archivo audiovisual al mismo. Este programa extrae el audio del audiovisual, sincroniza la voz con el texto, identifica al hablante y proporciona el fichero de subtítulos. Internamente, *ZarCaptions* utiliza una versión similar del reconocedor VivoLab que se emplea en este TFG. Este sistema se basa en la sincronización voz-texto utilizando una gramática forzada (detallada en el apartado 4.2.3 “Gramática forzada basada en estados finitos de perplejidad 1”), lo que se traduce en la no flexibilidad en la formación de frases, el reconocedor sigue fielmente el orden del texto presente en las transcripciones. Los subtítulos finales se generan en el orden en el que se encuentran en el texto proporcionado. En principio, dado que ya se dispone de las frases y el orden, solo es necesario sincronizarlos con la voz. Sin embargo, la calidad del audio puede no ser la óptima y pueden producirse retardos. Estos retardos pueden deberse a la aparición de canciones, ruidos o solapamiento entre hablantes, entre otros factores. Estos retardos pueden generar que, en el momento en que una frase o palabra se encuentre mal colocada en el tiempo, las frases que suceden a esta frase, también se encuentren desfasadas. El resultado no será el deseado. Además, puede conllevar una reacción en cadena de las frases sucesivas, incluso se podría llegar a un punto en el que el reconocedor se pierda por completo en la sincronización. Esta problemática lleva a la realización de este trabajo, con el fin de proporcionar cierta flexibilidad al reconocedor para la formación de frases, para que las frases puedan ser detectadas sin desfase y de este modo, evitar que el reconocedor se pierda. Después, se trata de corregir las zonas erróneas donde, debido a esta flexibilidad, el reconocedor ha colocado frases o palabras en momentos incorrectos.

1.5. Organización de la memoria

- **CAPÍTULO 1. Introducción**
  Contiene la motivación y los objetivos que han llevado a cabo a la resolución de este TFG. Breve introducción al campo del reconocimiento de voz y de la sincronización.
  También se narra el punto de partida y la problemática que surge, y se pretende solventar en este trabajo.
- **CAPÍTULO 2. Reconocedor automático del habla**
  Se introduce el procesamiento del habla y se detalla el funcionamiento del reconocedor automático del habla, así como los elementos que lo forman.
- **CAPÍTULO 3. Implementación de la sincronización de textos y los respectivos archivos audiovisuales**
  Este capítulo relata el diseño del sistema realizado. El procedimiento que se sigue hasta alcanzar el objetivo, el fichero de subtítulos.
- **CAPÍTULO 4. Herramientas integradas**
  Se describen las herramientas utilizadas y su papel en el proceso.
- **CAPÍTULO 5. Experimentación**
  Se muestran los resultados de las distintas pruebas. También se experimenta con programas con distintas dificultades y calidades de audio.
- **CAPÍTULO 6. Conclusiones**
  Se exponen las conclusiones y resultados del proceso.
- **CAPÍTULO 7. Bibliografía**
  Se nombran las fuentes empleadas para la obtención de la información.

CAPÍTULO 2. Reconocedor automático del habla

Dado que el reconocedor automático del habla es un elemento muy importante en este trabajo, es necesario explicar en qué consiste y que algoritmos utiliza. Para ello, se hace una breve introducción al RAH (reconocimiento automático del habla). Después, se trata cómo se lleva a cabo el procesamiento de la señal y se explica el modelo de canal ruidoso donde se interpreta porqué son necesarios el modelo acústico y el modelo de lenguaje en un reconocedor y en qué consisten los mismos. Por último, se expone el método empleado para decodificar incorporado en este reconocedor, algoritmo de Viterbi.

2.1. Sistemas de procesamiento del habla

En un sistema de procesamiento del habla, podemos distinguir tres áreas: **reconocimiento del habla**, **síntesis de voz** y **comprensión del lenguaje hablado**. En lo que concierna a este trabajo, este se centra en la primera área nombrada, el reconocimiento del habla.

2.1.2. Reconocimiento automático del habla

Extraer la información de la señal de voz no es proceso trivial, depende de múltiples factores tanto ambientales (posibles ruidos o interferencias) como lingüísticos (acento, género del hablante, etc.) que dificultan el proceso. Su función principal es realizar predicciones de manera que se obtenga la secuencia de palabras más probable partiendo del audio que se le provee el sistema de reconocimiento.

Desde un punto de vista más técnico, la idea principal de un sistema de reconocimiento del habla es conseguir convertir la señal de voz en texto. Para ello, se pueden distinguir distintos bloques dentro del sistema, véase la figura 1.

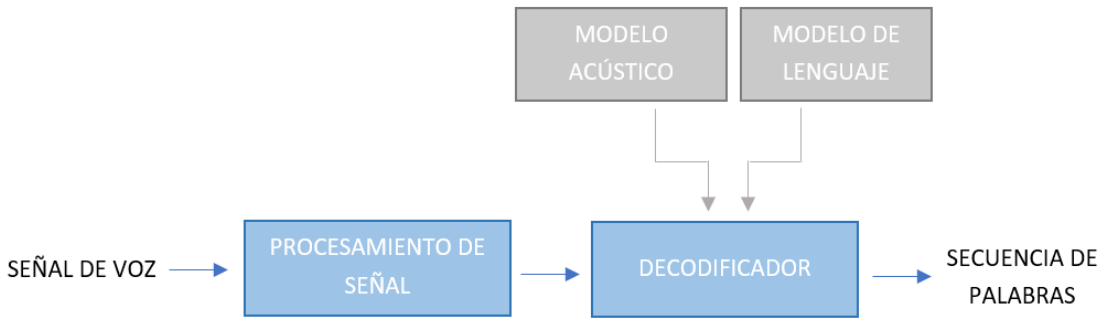

Figura 1. Sistema básico de reconocimiento del habla

En [1] se expone cómo la señal de voz, en primer lugar, debe ser procesada, cuya salida llega a un decodificador. Este emplea los modelos acústicos y de lenguaje para crear la secuencia de palabras más probable. El modelo acústico hace referencia a la información fonética, de género, de variabilidad del entorno, o información necesaria para la distinción entre hablantes, es decir, en general, conocimientos acústicos. El modelo de lenguaje se centra en las palabras, la relación entre las mismas y la probabilidad de formar una secuencia de palabras u otra. Es importante que los modelos puedan manejar las derivas del lenguaje debidas al acento del posible hablante, al vocabulario empleado, palabras más probables, dialecto, etc.

2.2. Procesamiento de la señal

El objetivo es analizar la señal de voz y generar una secuencia de parámetros relevantes para el reconocimiento de voz. Estos parámetros deben permitir distinguir entre distintos sonidos de una manera eficiente. Se suele trabajar en el dominio de la frecuencia.

El método más extendido es **Mel-frequency cepstral coeffients (MFCC)** [6]. Se trata de un análisis localizado y se trabaja con ventanas de 25 ms. Esos segmentos se consideran estacionarios por lo que es un tamaño conveniente y de ellos se extraen los coeficientes MFCC. También se aplica un desplazamiento de 10 ms entre ventanas. Concretamente cada una de estas tramas deben contener 39 coeficientes MFCC. Adicionalmente, la intención de obtener estos vectores característicos es ofrecer información sobre ruidos o identificación de hablante de una manera fiable.

El proceso a seguir para la extracción de los vectores anteriormente nombrados se muestra en la figura 2.

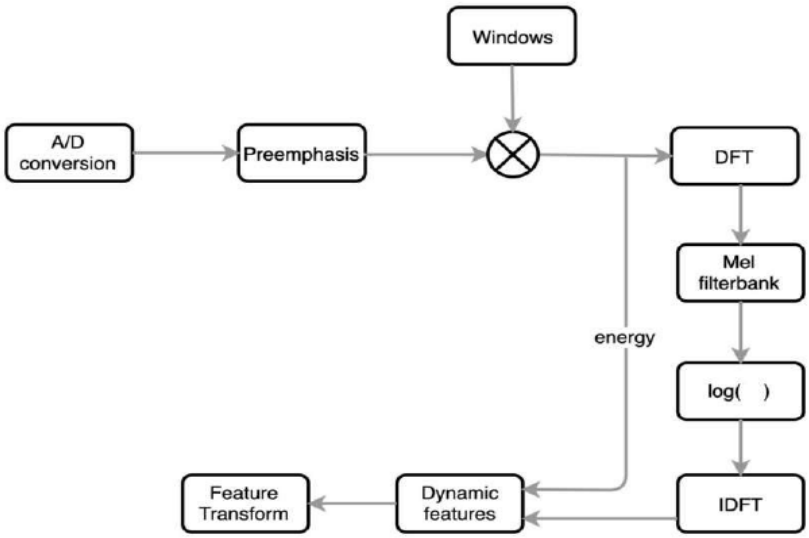

Figura 2. Método Mel-Frequency Cepstral Coefficients [6]

- **CONVERSOR A/D:**

La señal de voz se debe convertir en una señal digital para su posterior tratamiento. La frecuencia de muestro es de 16KHz.

- **PREÉNFASIS**

Este bloque se centra en la ecualización de la voz. Se aumenta la cantidad de energía en las altas frecuencias aplicando un filtro paso alto. De esta manera el modelo acústico podrá tratar de una mejor manera las componentes de alta frecuencia con el fin de compensar los efectos del pulso glotal, el cual es paso bajo.

- **ENVENTANADO**

Como ya se ha mencionado, se emplean ventanas de Hamming de 25 ms y 10 ms de desplazamiento entre tramas para no provocar un descenso brusco en la amplitud, lo que podría provocar ruido.

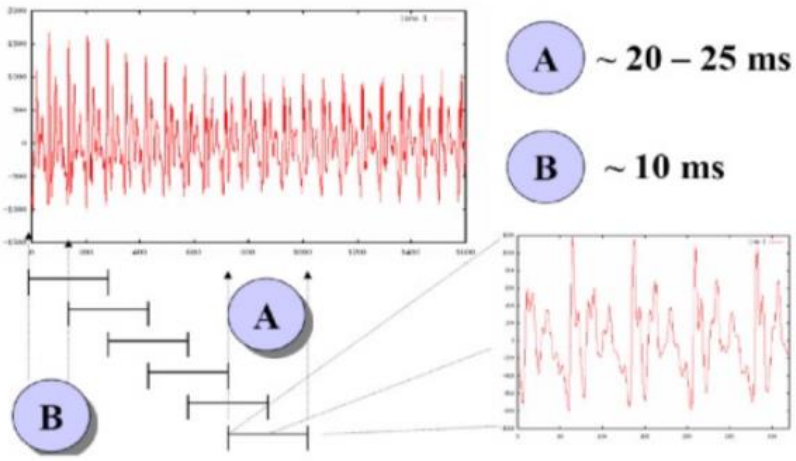

Figura 3. MFCC. Enventanado [6]

- **TRANSFORMADA DISCRETA DE FOURIOR (DFT)**

Se aplica la transformada discreta de Fourier para convertir la señal al dominio de la frecuencia:

$$X[k]= \sum_{n=0}^{N-1} x[n] \exp\left(-j \frac{2\pi}{N} kn\right) \quad (2.1)$$

- **BANCO DE FILTROS MEL**

La percepción auditiva en humanos no es lineal, sino logarítmica, lo que se traduce en mayores resoluciones a bajas frecuencias que a altas frecuencias. Para altas frecuencias, la sensibilidad del humano es menor. La escala Mel asigna las frecuencias que se perciben por un humano en términos de resolución de frecuencia.

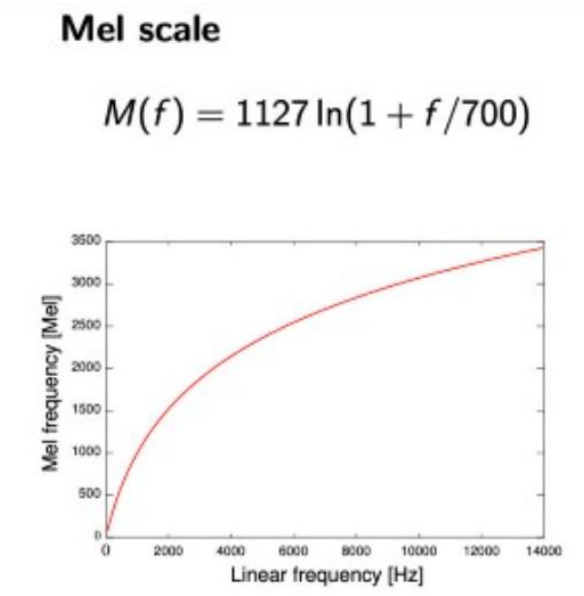

Figura 4. MFCC. Escala Mel [6]

En la práctica, se aplican filtros triangulares paso banda para imitar el oído humano, filtros no lineales.

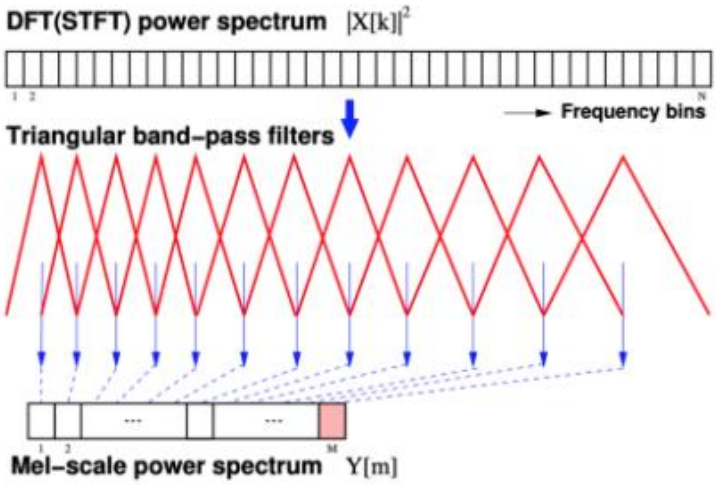

Figura 5. MFCC. Banco de filtros Mel [6]

Como resultado de aplicar el banco de filtros, se obtiene Y_t[m] como se muestra en la expresión 2.2. El proceso se denomina **Mel Binning**.

$$Y_t[\text{m}] = \sum_{k=1}^{N} W_m[k] \ |X_t[k]|^2 \qquad (2.2)$$

La componente $|X_t[k]|^2$ se trata de la señal x[n] tras aplicar la transformada DFT y elevarla al cuadrado, donde se obtiene el espectro de energía y $W_m[k]$ el banco de filtros. Y_t[m] hace referencia a la energía espectral en escala Mel de un segmento *m*, de manera que se representa la banda de frecuencias que se cubre en ese segmento. K simboliza los valores de la transformada DFT.

- **LOGARITMO**

El oído humano es menos sensible a pequeños cambios de energía en niveles de alta energía que a pequeños cambios en niveles de baja energía. En otras palabras, sigue un comportamiento logarítmico. Se debe aplicar el logaritmo a la salida tras aplicar el banco de filtros.

Por otro lado, se necesita eliminar la componente F0, frecuencia fundamental, que corresponde al Pitch, componente que varía en función del hablante, y garantizar que el resto de los coeficientes sean independientes entre sí.

- **CEPSTRUM - IDFT**

"Cepstrum" hace referencia a la inversa de las primeras 4 letras de "Spectrum". El espectro logarítmico contiene tanto la información de pitch como la información fonética necesaria para determinar qué se está diciendo. El siguiente paso sería conseguir separar estas dos componentes. Para ello se aplica la transformada inversa de Fourier (IDFT). Si se observa la figura 6, se puede apreciar que se distingue a simple vista la componente F0 de la información fonética. El pitch se traduce en un pico en el periodo de pitch, T.

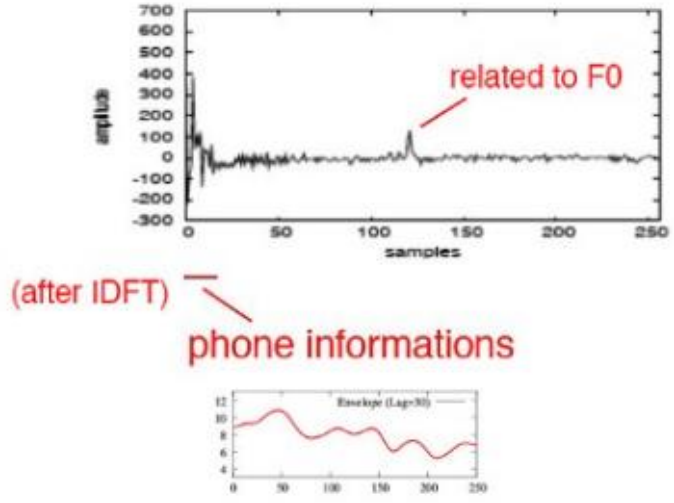

Figura 6. MFCC. Señal tras aplicar IDFT [6]

Solo se necesita analizar los coeficientes situados a la izquierda de la gráfica (Figura 6) y eliminar F0. Se toman los primeros 12 valores acústicos del Cepstrum. Dado que el espectro logarítmico es real y simétrico, se puede considerar que la IDFT es equivalente a realizar la **DCT (Discrete cosine transformation)**. Además, aplicar la DCT da lugar a componentes incorrelados, por lo que se cumple los dos requisitos anteriormente nombrados.

$$y_t[\mathrm{n}] = \sum_{m=0}^{M-1} \log(Y_t[m]) \cos\left(n(m+0.5)\frac{\pi}{M}\right) \quad (2.3)$$

- **COEFICIENTES DINÁMICOS (DELTA)**

Si MFCC contiene 39 coeficientes, partiendo de los 12 obtenidos en el paso anterior, se continúa con la adquisición del resto. El 13º parámetro es la energía de cada trama. El resto de los coeficientes hacen referencia a la evolución dinámica de los coeficientes acústicos, ofrecen información de contexto analizando de forma conjunta la trama previa y la inmediatamente próxima. Para obtener 13 de estos parámetros se computa d(t), la primera derivada.

$$\mathrm{d(t)} = \frac{c(t+1)-c(t-1)}{2} \quad (2.4)$$

Los 13 parámetros restantes se obtienen a través de la segunda derivada de c(t).

- **SUSTRACCIÓN DE LA MEDIA DEL CEPSTRUM**

El concepto clave es que en el dominio cepstral, las convoluciones se convierten en sumas. La sustracción de la media implica eliminar distorsiones convolutivas, ruidos convolutivos no variantes con el tiempo. Por ejemplo, el cambio de un micrófono o un ruido convolucional de una sala, en definitiva, las variaciones que puedan surgir a la hora de grabar la voz en determinadas condiciones ambientales.

$$\widehat{y_t}[\mathrm{j}] = y_t[\mathrm{j}] - \mu(\mathrm{y[j]}) \quad (2.5)$$

2.3. Modelo de canal ruidoso

La secuencia de palabras que un hablante tiene la intención de transmitir puede estar sometida a distorsiones debidas al ambiente u otras perturbaciones, por lo que la señal recibida por el decodificador podría no ser la deseada. Estas perturbaciones se catalogan como **canal ruidoso**.

Este canal ruidoso [2] puede ser tratado como una **inferencia Bayesiana**, tratar de determinar una palabra W, que se encuentra oculta, a través de una observación, X. Para obtener la palabra estimada $\widehat{W}$, debemos maximizar la probabilidad de obtener W dado X, es decir:

$$\widehat{W} = \underset{\widehat{W}}{\operatorname{argmax}}\, P(W \mid X) \tag{2.6}$$

Resolver esta expresión no es trivial por lo que hacemos uso de la siguiente equivalencia:

$$\widehat{W} = \underset{\widehat{W}}{\operatorname{argmax}}\, P(W \mid X) = \underset{\widehat{W}}{\operatorname{argmax}}\, \frac{P(X \mid W)\, P(W)}{P(X)} \tag{2.7}$$

Dado que P(X) no varía con cada palabra, podemos reducir la equivalencia anteriormente como se muestra en 2.8.

$$\widehat{W} = \underset{\widehat{W}}{\operatorname{argmax}}\, P(W \mid X) = \underset{\widehat{W}}{\operatorname{argmax}}\, P(X \mid W)\, P(W) \tag{2.8}$$

Una vez llegado a este punto, podemos definir $P(X \mid W)\ y\ P(W)$. $P(X \mid W)$ es el **modelo acústico**, y concretamente se emplean los **modelos ocultos de Markov**. P(W) es el **modelo de lenguaje**, el cual alternaremos para observar cual ofrece mejores resultados.

2.4. Modelo acústico: Modelos ocultos de Markov

El modelo acústico abarca la información acústica del reconocedor. Una palabra, fonema u sonido puede sonar de manera diferente al ser pronunciado. Es importante elegir las unidades acústicas adecuadas para llevar a cabo el modelado. Se empelan los fonemas contextuales [5]. Unidades que modelan estadísticamente los fonemas de la lengua con la que se trabaja, pero añadiendo rasgos de contexto, en otras palabras, información acerca del fonema que le precede o sucede.

Podemos ver estos fonemas o palabras como estados de un proceso de producción, donde se puede transitar entre ellos, aunque no todas son posibles ni cuentan con la misma probabilidad de transición. Un estado emite sonidos conforme a una función de densidad de probabilidad predefinida dado el proceso de producción de voz. Debido a estos sucesos, se utilizan los **modelos ocultos de Markov (HMM)** para modelarlos, sistema que permite modelar estas observaciones acústicas. Los modelos ocultos de Markov se explican detalladamente en el anexo A.

2.5. Algoritmo de Viterbi

Para poder obtener la secuencia más probable con un modelo como HMM, donde hay eventos ocultos, el reconocedor emplea para decodificar el **algoritmo de Viterbi** [2], un algoritmo de programación dinámica. La **programación dinámica** [3] se basa en el principio de optimalidad, la idea principal es que dado un estado inicial cualquiera, el sistema en cuestión debe ser capaz de obtener la solución más óptima conforme al estado actual en el que se encuentre, y así sucesivamente, hasta finalizar el proceso.

Aclarado el anterior concepto, en la figura 7 se observa un diagrama que representa el funcionamiento del algoritmo de Viterbi [2].

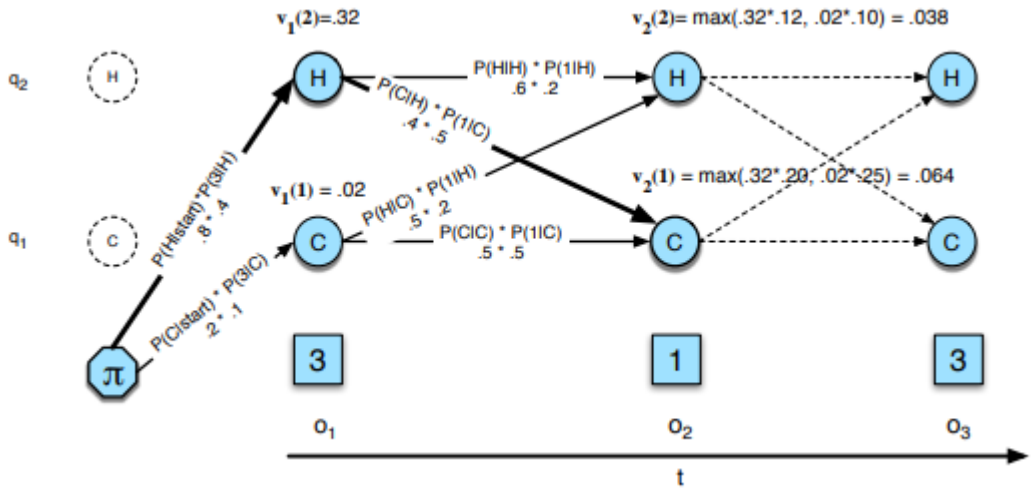

Figura 7. Esquema sobre el funcionamiento del algoritmo de Viterbi [2]

El estado π representa el estado inicial, del cual se parte. En el caso de este TFG, el estado inicial va a ser solamente uno, por lo que la probabilidad de este es 1 y 0 en el resto, aunque no siempre se sigue esta norma. Dado un modelo λ = (A,B), donde A y B son las probabilidades de transición y de observación, en cada estado, v_t(j) representa la probabilidad de que el HMM se encuentre en el estado j tras t observaciones y haber pasado por la secuencia más probable q_1, …q_{t-1} hasta el instante t-1, es decir:

$$v_t(j) = \max_{q_1, \ldots q_{t-1}} P(q_1, \ldots q_{t-1}, o_1, o_2 \ldots o_t, q_t = j \mid \lambda) \quad (2.9)$$

Cada estado se computa de manera recursiva, por lo que conociendo la probabilidad de haber estado en todos los estados para t-1, para un estado q_j en el instante t, se muestra el valor de v_t(j) en la expresión 2.10.

$$v_t(j) = max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t) \quad (2.10)$$

donde

- v_{t-1}(i) → Probabilidad de la secuencia de Viterbi anterior del instante de tiempo anterior.
- a_{ij} → Probabilidad de transición del estado i al estado j.
- $b_j(o_t)$ → Probabilidad de observación del estado del símbolo o_t dado el estado actual j.

Por último, resaltar que en este método se hace uso de "**backpointers**". Para calcular el resultado más óptimo en cada estado, se debe hacer analizar la ruta de estados ocultos que anteceden al estado actual.

2.6. Modelos del lenguaje

Los modelos de lenguaje determinan la probabilidad de aparición de palabras o frases, ofrece información léxica y sintáctica al sistema. Está altamente relacionada con el origen del hablante u otros factores que pueden modificar esta información. Estas condiciones implican que las secuencias de palabras estén dotadas de coherencia y lógica.

El vocabulario con el que se estructuran estas gramáticas es el propio de los programas a analizar. Se experimenta con las gramáticas. Además de experimentar con el tipo de gramática, se tratan dos formas de escoger este vocabulario que las forman. Dado que se estudian con detenimiento todas estas alternativas y se trata de un elemento clave de este TFG, se explican de manera detallada en el apartado 4.2 “Gramáticas”.

CAPÍTULO 3. Implementación de la sincronización de textos y los respectivos archivos audiovisuales

En primer lugar, en la figura 8 se muestra un esquema general del proceso. Seguidamente, se explica más detalladamente cada una de las partes.

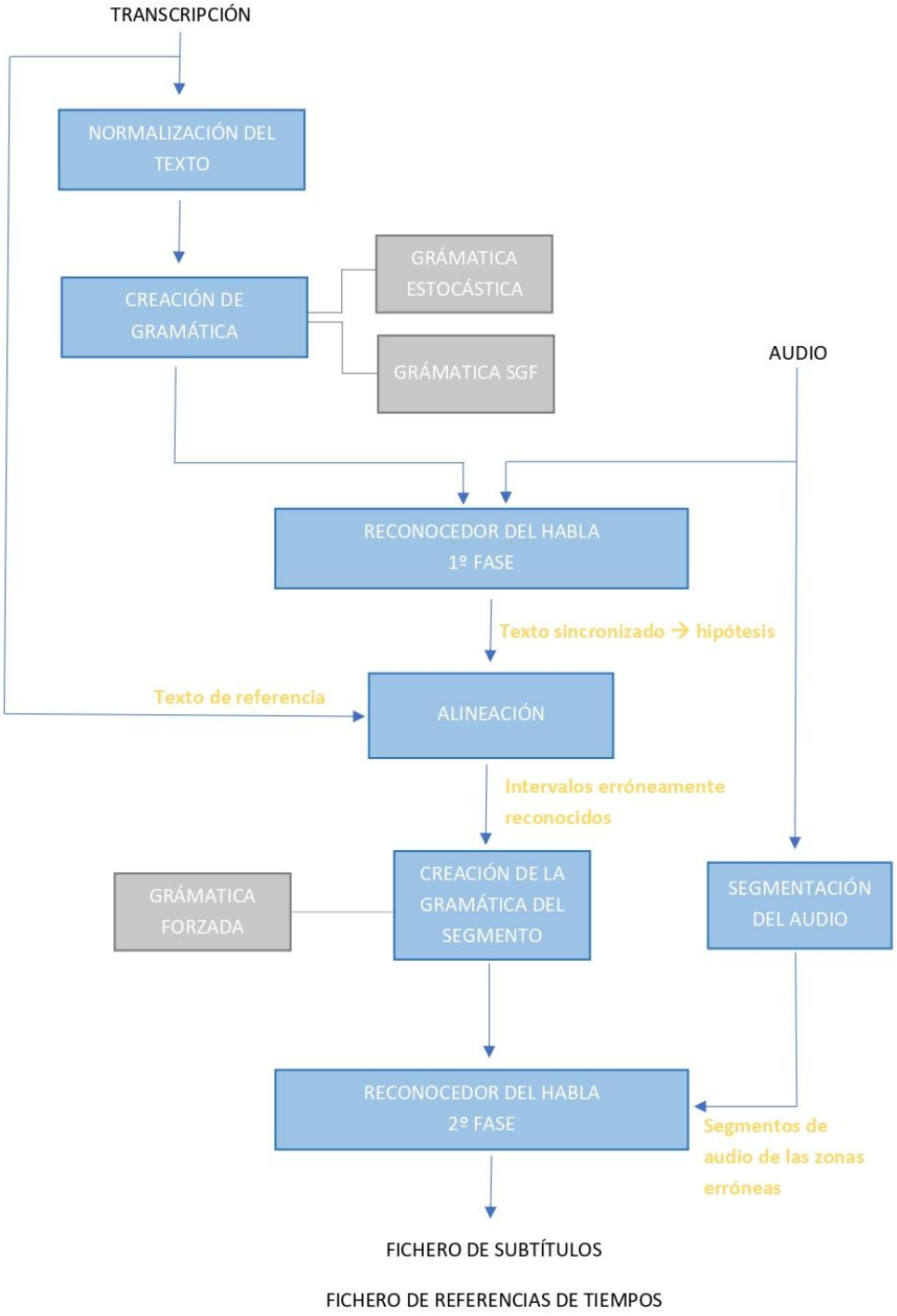

Figura 8. Esquema general del proceso

En el apartado 1.2 “Objetivo”, se ha explicado con gran brevedad cual es la finalidad de este TFG. En este capítulo se mostrará de manera más clara y extendida cómo se ha llevado a cabo.

Para explicar el proceso, se toma como ejemplo un solo archivo audiovisual y su transcripción, teniendo en cuenta que este mismo proceso podrá hacerse con cualquier otro programa. Para que el sistema que se desarrolla funcione correctamente es importante que las transcripciones a priori de las que se parte estén correctamente escritas y con los signos de puntuación correctamente colocados. En estos textos, y como ya se ha dicho en la introducción, al principio de cada fragmento de texto se indica el locutor, entre paréntesis. Este locutor puede tener un nombre concreto o puede ser referenciado mediante un número.

Se comienza modificando las transcripciones a un formato concreto que el reconocedor es capaz de leer. Esta modificación consiste en colocar “<ini>” y “<fin>” al comienzo y al final de cada línea del fichero original, los acentos como “á” o caracteres especiales como ”ü” o “ñ” deben ser sustituidos por “ ’a”, ”~u” y “~n” respectivamente. Además, los números, los signos como dólar ($) o euro (€) u otros deben ser escritos con letra, así como cambiar las abreviaciones por su palabra sin abreviar. En términos generales, **normalizar el texto.** En este caso no se deja constancia de los locutores. También debemos obtener el mismo texto normalizado, sin añadir “<ini>” y “<fin>” pero sí con el resto de las modificaciones y los hablantes, dado que la salida del reconocedor no cuenta con esta información y se necesitará añadir en el fichero de referencias de tiempos (extensión .stm).

Se sigue con la puesta a punto del sistema de reconocimiento con gramáticas que permiten cierta libertad en la formación de frases, para ello se proporciona el audio sin cabecera (audio con extensión. raw) y la gramática correspondiente. En esta primera fase, se prueban dos gramáticas, las cuales se tratan de manera más extensa en el apartado 4.2 “Gramáticas”. Lo importante para entender el proceso que se sigue es esa flexibilidad que proporciona la gramática al reconocedor para la creación las frases. Debido a esta libertad, las frases que se muestran en el fichero resultante del reconocedor no son necesariamente las frases dichas en el texto original. Por ello, será necesario detectar en el tiempo estas zonas que no se encuentran en el texto original para realizar una segunda fase de reconocimiento en ellas. Esta **primera fase de reconocimiento** es fundamental, ya que se puede tomar como puntos de referencia estas zonas que sí se han detectado correctamente en el tiempo, a partir de las cuales se reconocen las erróneas. La salida del reconocedor son las palabras que se detectan junto con los instantes de tiempo en el que el hablante dice la palabra.

Llegados a este punto, es necesario emplear una herramienta que nos permita hallar los fragmentos de texto que han sido detectados correctamente y las zonas de incertidumbre donde no se conocen las frases dichas por el hablante. Para ello, se integra *el toolkit SCTK (scoring toolkit) del NIST*, concretamente el programa *Sclite*, proporcionando al mismo el texto de referencia y el texto salida del reconocedor para que se encargue de la **alineación de palabras**. Este sistema se explica de manera más extensa en el apartado 4.1 “Programa Sclite”.

Observando varios ejemplos de la salida del programa *Sclite*, se toma la decisión de que los puntos son una fuente de información fiable, y que se detectan de una manera bastante leal al texto original. Por ello, se toman como referencia los puntos, a partir de los cuales se analizan las frases y se decide si son erróneas o no. A continuación, se hallan los intervalos que requieren resincronización, los cuales pueden contener varias frases.

Una vez conocemos los intervalos correctos, podemos tomar estos segmentos como **puntos de anclaje**. De esta manera, solo se vuelve a reconocer las zonas conflictivas en esta **segunda fase de resincronización**. El fichero que devuelve el programa *Sclite* solo contiene el texto como tal, no

contiene los tiempos, por lo que contando con el fichero resultante del programa *Sclite*, los tiempos de la salida del reconocedor y el texto normalizado donde constaban los hablantes, se genera un fichero de referencias de tiempo donde constan las zonas correctas y las erróneas, así como los instantes donde se hallan y los locutores correspondientes.

La siguiente fase de este TFG, trata de leer ese fichero de referencias obtenido en el paso anterior y volver a reconocer las zonas de incertidumbre. No es trivial marcar el instante de tiempo inicial y final de cada fragmento a reconocer, ya que, aunque la primera opción podría ser partir directamente de la primera frase mal detectada y acabar en la última frase errónea, no sería la opción más adecuada. Pueden surgir ciertos problemas, principalmente que, si la frase anterior correctamente detectada es muy corta, cabe la posibilidad de que no esté adecuadamente situada en el tiempo y que el reconocedor la haya situado en un instante de tiempo que no corresponde con dicha frase. Otro problema que puede surgir es que la frase anterior o posterior a una incorrecta, sí se empiece a detectar en el momento idóneo, pero se alarguen en el tiempo más de lo deseado debido a que el audio proporcionado no sea una señal limpia, de que el reconocedor confunda un trozo de canción con parte del texto, etc. por lo que el instante de tiempo donde debería empezar la frase incorrecta se encuentre desfasado. Debido a estos u otros problemas que puedan surgir, se decide forzar como incorrectas las frases anterior y posterior a la frase incorrecta, y en adición, si estas frases forzadas como erróneas son de longitud menor a cuatro palabras, se decide forzar la anterior/posterior a la misma, dado que podrían no ser unos puntos de anclaje fiables.

Una vez decididos cuales son estos puntos de anclaje a partir de los cuales el reconocedor debe volver a reconocer, **se proporciona al reconocedor todos los segmentos de audio respectivos a las secciones a analizar y sus correspondientes gramáticas**. En esta segunda fase de reconocimiento, la gramática es una gramática forzada basada en estados finitos de perplejidad 1, la cual se basa en forzar que el extracto reconocido sea idéntico al texto proporcionado a priori, es decir, no se permite la libertad que caracterizaba a la primera fase de reconocimiento. Esta gramática se explica de manera más extensa en el apartado 4.2 “Gramáticas”.

Una vez obtenida la salida del reconocedor, se vuelve a formar el fichero de referencias de tiempo y de subtítulos, uniendo los fragmentos resincronizados y los ficheros de subtítulos/referencias con errores.

Se escoge devolver como salida final del sistema el fichero de subtítulos de extensión .srt, porque es un formato ampliamente utilizado, el cual sigue ciertas normas para acoplar adecuadamente los subtítulos al archivo audiovisual, como mostrar un máximo de 39 caracteres por subtítulo o cortar una frase en el caso de que se superen los 39 caracteres. En adicción, se devuelve el fichero de referencias de tiempo porque, aunque no es útil a nivel de emisión de subtítulos debido a que los resultados se muestran por frases sin importar su longitud, lo cual no es práctico a la hora de visualizarlo simultáneamente con el programa, sí es de gran utilizar cuando se trabaja con el reconocedor y se quieren visualizar los resultados en ese preciso instante. Mientras que el fichero de subtítulos se devuelve con los signos de puntuación y acentos correctamente colocados nuevamente, el fichero de referencias aún se encuentra normalizado por la misma razón anteriormente expuesta. La estructura de los dos ficheros salida se muestra en el anexo B.

Una vez expuesto el proceso que se sigue en este TFG, se puede comprender con mayor facilidad en que consiste este TFG y por qué no utilizar gramáticas forzadas exclusivamente, y por lo tanto obligar desde un principio que todas las frases reconocidas por el reconocedor sean directamente las mismas que aparecen en el texto original y sigan el mismo orden que en este, tal y como se muestra en el apartado 1.3 “Estado del arte”.

CAPÍTULO 4. Herramientas integradas

4.1. Programa *Sclite*

El programa *Sclite* se aplica para la **alineación de textos**. *Sclite* necesita como parámetros de entrada el texto de referencia y el texto hipótesis donde la hipótesis es la salida del reconocedor sin los tiempos, solo el texto. El programa se basa en el método **DTW (dynamic time warping)** que se describe en el anexo C.

Como salida se devuelven tres archivos, el primero de ellos contiene la alineación de los textos en sí, donde se muestran tres partes diferenciadas, la referencia, la hipótesis y la evaluación. La evaluación está compuesta de tres componentes: Borrados, inserciones y sustituciones. En caso de que una palabra esté contenida en la referencia pero no en la hipótesis, se trata de un borrado. En el caso contrario, donde la palabra se encuentra solamente en el texto hipótesis, implica una inserción. En cualquiera de los dos sucesos anteriores, la ausencia de una palabra se refleja con asteriscos en la línea correspondiente a la referencia o la hipótesis. Como última opción, se puede dar la sustitución cuando la palabra de referencia y la reconocida no son la misma. Dado que el reconocedor también da como salida signos de puntuación, el punto y la coma, también se tratan de la misma manera que las palabras. Además, cabe destacar que los errores se muestran con las palabras en mayúsculas y las palabras detectadas de manera idónea se muestran en minúsculas.

En la figura 9 se muestra un ejemplo de alineación de textos utilizando este programa. La primera línea se trata de la referencia, la segunda línea de la hipótesis y la última de la evaluación. Los borrados se traducen con la letra D, las inserciones con la letra I y las sustituciones con la letra S.

```
HAY QUE BUSCARLO ,  S'I ,  S'I  . y suerte , violeta , con el reto . a por 'el , 'animo * ** . GRACIAS .
*** *** ******** NO .   EH CIEN . y suerte , violeta , con el reto . a por 'el , 'animo . NO . AH      .
D   D   D        S  S   S  S                                                              I I    S
```

Figura 9. Alineación de textos con el programa Sclite

En lo que concierna a este TFG, solo se tienen en cuenta los borrados y las sustituciones, las inserciones se omiten a la hora de elaborar los fragmentos incorrectos que habrá que volver a sincronizar. Esto último se debe a que las inserciones, dado que no se encuentran en el texto de referencia, no van a incluirse en el fichero final. Además, las inserciones pueden deberse a que el reconocedor interprete un trozo de canción como parte del texto u otros factores, por lo que no será necesario tratar con las mismas.

Los otros dos ficheros de salida muestran los resultados de la alineación. Uno de ellos muestra los porcentajes de errores totales y errores debidos a sustituciones, borrados e inserciones de manera individual. El otro fichero de salida muestra estos mismos resultados en términos absolutos, es decir, muestra el número de palabras totales y la distribución de palabras correctas e incorrectas que corresponda. Es interesante conocer estos datos para el capítulo de experimentación, donde estos datos se emplean para realizar estudios sobre el funcionamiento de las diferentes gramáticas utilizadas en la primera fase de reconocimiento y de los diferentes programas de RTVE que tienen una dificultad de reconocimiento diferente, por lo que los resultados del programa *Sclite* pueden variar en función de estos factores.

4.2. Gramáticas

Como se ha explicado de manera abreviada en el capítulo 3 "Implementación de la sincronización de textos y los respectivos archivos audiovisuales", se hace uso del reconocedor en dos ocasiones diferentes. En la primera fase, se requiere dotar de cierta flexibilidad al reconocedor para la formación de frases, por lo que se experimenta con **gramáticas estocásticas** y **gramáticas basada en estados finitos (FSG)**. En la segunda fase, se requiere forzar las frases que deben aparecer en el segmento erróneo, por lo que solamente se trabaja una gramática, **gramática forzada basada en estados finitos de perplejidad 1**.

En esta primera vez que se reconoce el archivo audiovisual, tanto para la gramática de estados finitos como para la gramática estocástica, se emplean dos tipos de vocabulario para crearlos y experimentar, con la intención de que se consigan los mejores resultados. Uno de estos vocabularios solamente está compuesto por la transcripción propia del archivo audiovisual. En otras palabras, la gramática sólo se forma con la información de un diccionario compuesto solo por las frases, palabras y expresiones que se van a pronunciar en el audiovisual. De este modo, la gramática de cada audiovisual a tratar es **única** y es necesario integrar una diferente para cada objeto de estudio. El otro tipo de vocabulario está compuesto por todas las transcripciones de un tipo de programa concreto. No se considera una gramática compuesta por todas las transcripciones de todos los diferentes programas analizados porque cada programa trata de temas diferentes y por lo tanto el vocabulario y expresiones podrían ser opuestos, por lo que los resultados serán considerablemente peores. Sin embargo, sí podría resultar interesante analizar una gramática **común para el mismo tipo de programa**. En principio, el vocabulario que aparece en cada programa del mismo tipo sí tendrá similitudes entre sí y puede proporcionar al reconocedor información acerca de las probabilidades de aparición de palabras, frases o expresiones y poder detectar con mayor facilidad las más probables. Por lo tanto, en el capítulo de experimentación se mostrarán los resultados de aplicar las distintas gramáticas y las diferentes maneras de formarlas.

4.2.1. Gramática estocástica

La **gramática estocástica**, se basa en aplicar probabilidades a las reglas propias del lenguaje. Siguiendo las propias reglas establecidas por el lenguaje, le añadimos un componente probabilístico. Este componente añade información acerca de las secuencias de palabras más probables de aparición. Dado que en este TFG se estudia la sincronización de programas de los que se dispone de sus transcripciones, se conoce el vocabulario que aparecerá en el audiovisual, por lo que obtener la información a priori acerca de las formaciones de secuencias más probables es relativamente sencillo y se puede emplear esta información para formar la gramática. Para modelar la gramática, se utiliza la herramienta *SRILM* de libre distribución. *SRILM* [4] es una herramienta para la construcción y aplicación de modelos de lenguaje estadístico, principalmente para reconocimiento de voz, etiquetado estadístico y segmentación, y traducción automatizada. Esta tecnología se basa en los modelos de lenguaje N-gramas, concretamente trigramas, concepto que se explica a continuación.

4.2.1.1. Modelos de lenguaje N-Gramas

Un n-grama es una secuencia de N palabras. El reconocedor VivoLab admite trigramas, secuencia de tres palabras, por lo que en la práctica será el n-grama empleado. Este apartado se basa en el capítulo 3 de [2]. En este capítulo se narra que para explicar el concepto de n-gramas se debe conocer P(w|h), la probabilidad de obtener la palabra *w* dado un historial *h,* donde *h* es una secuencia de palabras que

precede a *w*. Para verlo de una manera más sencilla se toma *h* como "El agua es tan transparente que" y *w* como la palabra "la". La probabilidad P(w|h) en este caso se puede ver en la ecuación 4.1.

$$\mathrm{P}(\text{la} \mid \text{El agua es tan transparente que}) = \frac{C(El\ agua\ es \tan transparente\ que\ la)}{C(El\ agua\ es \tan transparente\ que)} \quad (4.1)$$

Donde C(x) es el número de ocasiones que se ha observado ese orden de secuencia concreto. Cuanto mayor sea el corpus estudiado, mejor serán los resultados. En este caso, el corpus no es excesivamente grande, pero no es un problema ya que las secuencias de palabras que se buscan siempre estarán contenidas en este. Por otro lado, es necesario conocer la probabilidad de que, dada una secuencia de 6 palabras consecutivas, esta sea "El agua es tan transparente que", por lo que habría que volver a calcular el número de veces que se ha observado "El agua es tan transparente que" en las secuencias de 6 palabras, lo cual son considerables posibilidades. Bajo estas condiciones, resulta más sencillo calcular P(w|h) de la forma que se muestra a continuación.

La probabilidad de una secuencia de N palabras, $\mathrm{P}(w_1, w_2, \dots w_n)$, se calcula en la ecuación 4.2 y se considera que $\mathrm{P}(w_1, w_2, \dots w_n)$=$\mathrm{P}(w_1^n)$ por simplificación.

$$\mathrm{P}(w_1^n) = \mathrm{P}(w_1)\ \mathrm{P}(w_2|w_1)\ \mathrm{P}(w_3|w_1^2)\ \dots \mathrm{P}(w_n|w_1^{n-1}) = \prod_{k=1}^{n} \mathrm{P}(w_k|w_1^{k-1}) \quad (4.2)$$

La anterior expresión muestra la relación entre la probabilidad de una secuencia entera de palabras y la probabilidad condicional de una palabra dadas las palabras que anteceden a la misma. No siempre se va a conocer la probabilidad exacta de la secuencia entera, pueden aparecer secuencias de las que no se conoce su probabilidad. Por ello, en los modelos de lenguaje n-gramas **se estima esta probabilidad a partir de las últimas n-1 palabras**, es decir:

$$\mathrm{P}(w_n|w_1^{n-1}) \approx \mathrm{P}(w_n|w_{n-N+1}^{n-1}) \quad (4.3)$$

De esta forma, la probabilidad se aproxima de la siguiente manera:

$$\mathrm{P}(w_1^n) \approx \prod_{k=1}^{n} \mathrm{P}(w_k|w_{k-N+1}^{k-1}) \quad (4.4)$$

Donde $\mathrm{P}(w_k|w_{k-N+1}^{k-1})$ es la probabilidad de una palabra dadas las n-1 anteriores. En este caso, N=3 trigramas.

4.2.2. Gramática basada en estados finitos (FSG)

La gramática de estados finitos en términos generales permite menos flexibilidad para la formación de frases que en el caso de gramáticas estocásticas. Es un sistema compuesto por estados, donde cada estado es una palabra o signo de puntuación. Se puede transitar de un estado a otro añadiendo una probabilidad a esa transición. Lo interesante es conocer cómo se da ese tránsito entre estados. Siempre se parte de un estado inicial y se termina en un estado final. Podría haber varios estados iniciales y finales pero, en este caso, en cada frase del fichero de transcripciones se fuerza a que comience por "<ini>" y termine por "<fin>". Tras el estado inicial, la sucesión de estados no se centra únicamente en reglas probabilísticas basadas en la aparición de una palabra dadas las n-1 anteriores, sino que solo se pueden formar frases que ya se encontraban en el texto original, es decir, solamente se puede transitar por estados concretos que den como resultado una frase conocida. Por esta razón, las oraciones elaboradas tendrán lógica por sí mismas y darán lugar a frases coherentes. Sin embargo, no significa que la frase sea coherente con las frases que le preceden o le suceden.

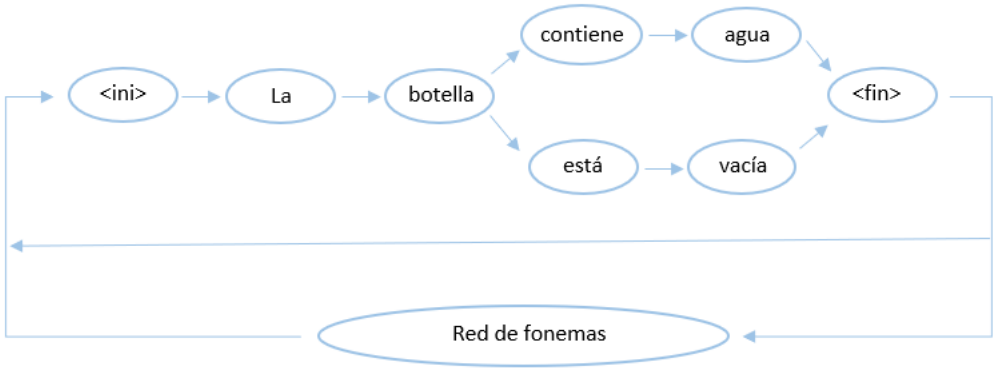

Figura 10. Esquema de estados finitos

En la figura 10 se muestra un esquema sencillo de estados finitos. Como se puede observar, aunque existen varios caminos, ambos son conocidos. A pesar de que hay varias posibles rutas fijadas, una vez nos encontremos en un estado que puede transitar a otros, está transición será probabilística. El estado final tiene como sucesores el estado inicial y una red de fonemas, la cual tiene a su vez como sucesor el estado inicial. La red de fonemas contiene todos los fonemas del idioma con el que se trabaja. Los sonidos detectados por este bloque no se muestran como salida del reconocedor. Su función es detectar posibles ruidos, canciones u otros sonidos que podrían surgir en el audio, pero no se desean reconocer, ya que no se encuentran en las transcripciones. De esta manera, una vez que se ha terminado de reconocer una frase y no se detecta la siguiente, se puede transitar a la red de fonemas hasta que se reconozca la próxima frase potencial. Esta movilidad entre estados se refleja con la perplejidad. La perplejidad podría definirse como el factor de ramificación medio. El mínimo valor que puede tomar es 1, y se puede incrementar indefinidamente. Bajo las condiciones de este TFG, su valor con esta gramática suele ser cercano a 1, lo que implica que, aunque se permite cierta flexibilidad, no es muy alta.

4.2.3. Gramática forzada basada en estados finitos de perplejidad 1

En la segunda fase de reconocimiento, se intenta corregir los segmentos detectados erróneamente, como ya se ha mencionado, y se decide utilizar una gramática forzada. Teniendo en cuenta que conocemos el texto que se va a pronunciar por los locutores y que, llegados a este punto, también se cuenta con unos puntos de anclaje fiables, se trata de volver a reconocer las zonas entre estos puntos. Punto de anclaje hace referencia a la frase anterior y posterior correctamente detectada y sus correspondientes tiempos, por lo que al reconocedor se le proporciona la gramática forzada formada por las frases a reconocer y el extracto de audio a analizar.

Como el propio nombre indica, se fuerzan las frases que el reconocedor debe dar como salida y el orden de las mismas. El reconocedor solo debe situarlas en el tiempo. Este tipo de gramática se puede considerar como una gramática basada en estados finitos. En esta fase, la perplejidad valdrá 1, ya que no se permite flexibilidad entre frases, su orden está fijado.

Una buena medida para comparar las tres gramáticas podría ser la perplejidad, ya que nos indica el numero promedio de posibles sucesores. En el capítulo de experimentación también se mostrarán estos resultados.

4.3. Base de datos

La base de datos utilizada son los programas de RTVE (Radio Televisión Española). Más específicamente se estudian los siguientes programas:

- Informativo 20H
- Al filo de lo imposible
- Arranca en verde
- Asuntos públicos
- Comando Actualidad
- Dicho y hecho
- España en comunidad
- Latinoamérica en 24H
- La mañana
- La noche en 24H
- La tarde en 24H Tertulia
- Millennium
- Saber y ganar

Para poder mostrar y entender los resultados de la experimentación, se va a hacer un breve resumen de la calidad acústica, la duración aproximada y la temática de cada uno de ellos, de manera que se entenderán mejor las dificultades de trabajar con estos programas.

Programa	Duración aproximada	Descripción
Informativo 20H	30-40 minutos	Noticias del día. Buena calidad de audio en términos generales.
Al filo de lo imposible	30 minutos	Documentales de deportes de riesgo al aire libre. Calidad pobre en actividades al aire libre.
Arranca en verde	30 minutos	Concurso dedicado a la seguridad vial. Buena calidad en general.
Asuntos públicos	120-150 minutos	Análisis de las noticias del día y transmisión en vivo de los eventos informativos más destacados. Buena calidad con algún solapamiento entre hablantes.
Comando actualidad	60 minutos	Espectáculo que presenta un tema actual de la mano de varios reporteros. Mala calidad en su mayoría, bastante solapamiento.
Dicho y hecho	120 minutos	Programa donde un grupo de 6 comediantes y celebridades compiten entre sí a través de divertidos desafíos. Considerables solapamientos entre hablantes e inflexiones del habla.
La mañana	90-120 minutos	Magacín de las mañanas, con una variada oferta de contenidos para toda la familia y con clara vocación de servicio público. Solapamientos entre hablantes e inflexiones del habla.

La tarde en 24H Tertulia	60 minutos	Tertulia sobre noticias políticas y económicas. Buena calidad en general, pocos solapamientos entre hablantes.
Latinoamérica en 24H	30 minutos	Programa sobre análisis e información enfocado en Iberoamérica. Buena calidad en general con predominante acento latinoamericano.
Millennium	60 minutos	Debate de ideas con la intención de resultar útil a los espectadores, acompañándolos en el análisis de acontecimientos cotidianos. Buena calidad en general, discurso tranquilo.
Saber y Ganar	40-50 minutos	Concurso diario que tiene como objetivo difundir la cultura de una manera entretenida. Tres concursantes demuestran su conocimiento y agilidad mental, a través de un conjunto de preguntas generales. Bueno calidad en general.
La noche en 24H	120 minutos	Programa de entrevistas con análisis para comprender lo que ha sucedido a lo largo del día. Contiene entrevistas con algunos de los protagonistas del día. Buena calidad en general con algún solapamiento entre hablantes.
España en comunidad	20-30 minutos	Ofrece informes detallados e información actual sobre las diferentes comunidades autónomas españolas. Buena calidad en general.

Tabla 1. Programas RTVE: Temática, duración y calidad acústica [5]

CAPÍTULO 5. Experimentación

Como se indica en el apartado 4.2 “Gramáticas”, experimentar con diferentes gramáticas es un punto clave en este trabajo, por lo que en este capítulo se muestran los resultados obtenidos al utilizar diferentes gramáticas en una primera fase de reconocimiento y se compara entre ellas. Además, se diferencia el método de formación de la gramática solamente con el texto perteneciente al archivo audiovisual estudiado y la gramática formada por todos los textos pertenecientes al mismo programa.

Una vez se ha obtenido la hipótesis en la primera fase, se identifican las palabras detectadas correcta y erróneamente, por lo que se mostrará el porcentaje de error en función del programa, de la gramática y de los distintos tipos de errores que pueden aparecer. Por otro lado, a partir de esas palabras, se forman los segmentos que hay que volver a reconocer, por lo que también se analiza los segmentos erróneos y su longitud en función del tipo de programa.

En el apartado 4.3 “Base de datos” se muestran las características de los distintos programas que se emplean, por lo que será interesante observar el tipo de audio que contiene ese programa y los errores que se producen en cada caso.

Una vez explicada de manera general la finalidad de este capítulo, se comienza con el estudio de la perplejidad de cada una de las gramáticas. Se recuerda que en esta primera fase se permite cierta flexibilidad en la formación de frases, por lo que en la perplejidad correspondiente a la gramática estocástica y a la gramática basada en estados finitos se refleja esta movilidad entre palabras. En el caso de la gramática forzada, aplicada en la segunda fase de reconocimiento, la perplejidad es 1 en todos los casos. En la tabla 2 se expone la perplejidad media por programa y por gramática. También se muestra el número de programas con el que se ha realizado la media.

PROGRAMA	**Informativo 20H**	**Al filo de lo imposible**	**Asuntos públicos**
Nº de programas	15	9	4
Perplejidad gram. estados finitos	1,01444	1,01176	1,01423
Perplejidad gram. estocástica	29,99773	32,55303	34,48362
Perplejidad gram. forzada	1	1	1
PROGRAMA	**Arranca en verde**	**Comando actualidad**	**Dicho y hecho**
Nº de programas	2	8	1
Perplejidad gram. estados finitos	1,05749	1,06969	1,07280
Perplejidad gram. estocástica	28,64346	42,33808	45,86652
Perplejidad gram. forzada	1	1	1
PROGRAMA	**España en comunidad**	**Latinoamérica en 24H**	**La mañana**
Nº de programas	22	8	4
Perplejidad gram. estados finitos	1,02242	1,01625	1,03447
Perplejidad gram. estocástica	29,15134	36,13310	41,63194
Perplejidad gram. forzada	1	1	1
PROGRAMA	**La noche en 24H**	**La tarde en 24H Tertulia**	**Millennium**
Nº de programas	18	9	8
Perplejidad gram. estados finitos	1,02682	1,02127	1,01587

Perplejidad gram. estocástica	37,09700	31,97403	33,13969
Perplejidad gram. forzada	1	1	1
PROGRAMA	**Saber y ganar**		
Nº de programas	4		
Perplejidad gram. estados finitos	1,05186		
Perplejidad gram. estocástica	26,15621		
Perplejidad gram. forzada	1		

Tabla 2. Perplejidad en función de las distintas gramáticas y programas

La gramática forzada siempre es 1, lo cual ha sido premeditado con el objetivo de determinar con exactitud y con la colocación deseada las palabras a reconocer en la segunda fase. No se puede analizar más información de la tabla con lo que respecta a esta gramática. En cambio, la comparación entre perplejidades correspondientes a la gramática estocástica y a la gramática basada en estados finitos sí resulta interesante, ya que a continuación se escoge entre ambas para emplear dicho formato en la primera fase. Hay una gran diferencia entre ambas, principalmente la gramática estocástica permite mayor libertad en la formación de frases, por lo que habrá más variedad de oraciones y no siempre formarán parte del texto original. La gramática de estados finitos es muy cercana a 1, lo que implica que, aunque se permite una cierta tolerancia en la creación de oraciones, será a un nivel mucho más bajo y sí pertenecerán al texto original. No se detectan cambios excesivamente bruscos entre perplejidades con distintos programas pero una misma gramática.

Se sigue con la comparación de la gramática estocástica y la gramática basada en estados finitos. La comparación se centra en el número de palabras erróneas en la primera fase de reconocimiento. Los resultados se obtienen gracias a la herramienta *Sclite*, con la que se realiza la alineación. Se muestran en porcentaje. Para hacer un análisis más global, se hace una media entre todos los programas de la misma serie. En estas tablas, se indica el número de programas con los que se ha hecho el promedio, el porcentaje de palabras correctas, el número de sustituciones, el número de borrados y el número de inserciones, y, por último, el error total (sustituciones, borrados e inserciones).

GRAMÁTICA ESTOCÁSTICA			
PROGRAMA	**Informativo 20H**	**Al filo de lo imposible**	**Asuntos públicos**
Nº de programas	15	9	3
Palabras correctas	51,802%	39,843%	37,166%
Sustituciones	42,895%	57,102%	58,105%
Borrados	5,303%	3,055%	4,729%
Inserciones	32,882%	76,748%	72,884%
Error total	81,080%	136,904%	135,719%
PROGRAMA	**Arranca en verde**	**Comando actualidad**	**Dicho y hecho**
Nº de programas	2	8	1
Palabras correctas	28,035%	25,232%	29,247%
Sustituciones	65,592%	70,432%	65,971%
Borrados	6,374%	4,336%	4,782%
Inserciones	26,940%	39,815%	48,903%
Error total	98,905%	114,583%	119,656%
PROGRAMA	**España en comunidad**	**Latinoamérica en 24H**	**La mañana**
Nº de programas	22	8	3
Palabras correctas	55,505%	52,279%	29,210%

Sustituciones	38,654%	43,765%	65,172%
Borrados	5,842%	3,956%	5,618%
Inserciones	30,384%	43,983%	27,750%
Error total	74,879%	91,703%	98,541%
PROGRAMA	**La noche en 24H**	**La tarde en 24H Tertulia**	**Millennium**
Nº de programas	16	9	8
Palabras correctas	29,189%	29,666%	51,732%
Sustituciones	64,746%	65,141%	41,377%
Borrados	6,065%	5,193%	6,892%
Inserciones	30,505%	34,129%	30,111%
Error total	101,317%	104,464%	78,379%
PROGRAMA	**Saber y ganar**		
Nº de programas	4		
Palabras correctas	50,412%		
Sustituciones	42,973%		
Borrados	6,615%		
Inserciones	29,712%		
Error total	79,300%		

Tabla 3. Porcentaje de palabras erróneas. Gramática estocástica. 1º Fase

GRAMÁTICA BASADA EN ESTADOS FINITOS			
PROGRAMA	**Informativo 20H**	**Al filo de lo imposible**	**Asuntos públicos**
Nº de programas	15	9	4
Palabras correctas	97,955%	85,530%	94,961%
Sustituciones	1,480%	11,659%	3,331%
Borrados	0,565%	2,811%	1,709%
Inserciones	2,124%	6,779%	16,663%
Error total	4,169%	21,249%	21,702%
PROGRAMA	**Arranca en verde**	**Comando actualidad**	**Dicho y hecho**
Nº de programas	2	8	1
Palabras correctas	73,279%	76,384%	77,091%
Sustituciones	23,224%	20,243%	21,062%
Borrados	3,497%	3,373%	1,847%
Inserciones	15,118%	14,954%	41,346%
Error total	41,838%	38,570%	64,256%
PROGRAMA	**España en comunidad**	**Latinoamérica en 24H**	**La mañana**
Nº de programas	22	8	4
Palabras correctas	96,610%	98,681%	89,507%
Sustituciones	2,560%	0,755%	7,740%
Borrados	0,830%	0,564%	2,753%
Inserciones	4,505%	2,761%	4,796%
Error total	7,895%	4,079%	15,289%
PROGRAMA	**La noche en 24H**	**La tarde en 24H Tertulia**	**Millennium**
Nº de programas	18	9	8
Palabras correctas	84,052%	79,436%	98,569%
Sustituciones	11,879%	14,828%	0,699%
Borrados	4,070%	5,736%	0,732%

Inserciones	4,469%	3,152%	1,577%
Error total	20,417%	23,716%	3,008%
PROGRAMA	**Saber y ganar**		
Nº de programas	4		
Palabras correctas	98,305%		
Sustituciones	0,888%		
Borrados	0,807%		
Inserciones	3,952%		
Error total	5,647%		

Tabla 4. Porcentaje de palabras erróneas. Gramática basada en estados finitos. 1º Fase

Primeramente, hay comentar que, aunque no se muestren los resultados de manera explícita, se ha tenido en consideración la formación de gramáticas SFG y estocásticas con todas las transcripciones pertenecientes a la misma serie de programas, y tras comparar los resultados con los obtenidos a través de las gramáticas creadas con las transcripciones individuales de cada fichero audiovisual se considera que una gramática conjunta es computacionalmente más costosa y los resultados son peores, por lo que **se desecha la opción de una gramática común** para ambos casos.

Para el análisis de las tablas 3 y 4, hay que tener en cuenta que, aunque sí son interesantes los errores debidos a las inserciones, no resultan de gran utilidad para la resolución de este TFG, ya que estos errores no se contemplan, se omiten. Se observa que las tasas de error varían de manera notoria según el programa a tratar. Tomando en consideración la tabla 1 "Programas RTVE: Temática y calidad acústica", los programas con una calidad acústica inferior dan peores resultados de manera general. Por poner un par de ejemplos, "Comando actualidad" o "Dicho y hecho" son unos programas con una calidad acústica más pesimista y queda reflejado en las tasas de error. Por contraposición, "Informativo 20H" o "Millennium" cuentan con una buena calidad acústica y también se refleja en los porcentajes de error.

Echando un rápido vistazo a la tabla de gramática FSG, llama la atención que "Arranca en verde" tiene una tasa de error más alta de lo esperable dada la definición acústica del audiovisual. Por esta razón se ha mirado los errores de manera individual. Se cuenta con dos programas donde los errores individuales son los siguientes:

Programa 1	Sustituciones	10,2 %
	Borrados	2,4 %
	Inserciones	14,8 %
	Error total	**27,4 %**
Programa 2	Sustituciones	36,6 %
	Borrados	4,6 %
	Inserciones	15,5 %
	Error total	**56,7 %**

Tabla 5. Porcentaje de palabras erróneas. Arranca en verde. Gramática FSG

La tasa de error del primer programa estaría más en concordancia con lo esperado. Sin embargo, el segundo programa incrementa considerablemente la tasa de error[1]. Hay que tener en cuenta que solo se cuenta con dos programas para realizar el promedio, podría ocurrir y ocurre, que el promediado

[1] Las tasas de error totales, incluidas en la tabla 4, podrían no ser exactamente el promedio de los porcentajes mostrados en la tabla 5 ya que la tasa de error total se obtiene teniendo en cuenta el número de palabras de cada transcripción. Los porcentajes de la tabla 5 son individuales de cada programa.

sea peor en comparación el promediado obtenido con un mayor número de programas de la misma serie.

Una vez analizados estos resultados, **se llega a la conclusión de que la gramática basada en estados finitos da mejores resultados**, tasas de errores más bajas. La diferencia es considerable por lo que la **gramática estocástica se descarta** y se trabaja solamente con la primera.

Para concluir, también se van a tratar el número de segmentos erróneos que se deben reconocer en la segunda fase en función del programa y su longitud por palabras. Los segmentos que se vuelven a resincronizar en la segunda fase incluyen frases al principio y al final, que, en principio, están bien sincronizadas, pero que se incluyen en el segmento para garantizar una buena sincronización. Hay que destacar que, durante la experimentación, han surgido ficheros donde el número de frases erróneas se prolongaban más allá de la frase anterior y posterior supuestamente bien detectadas. Este hecho se debe a que, aunque las frases colocadas sí son correctas y, por tanto, el programa de alineación da por buenas esas frases, no se han colocado de manera adecuada en el tiempo. El programa *Sclite* solo compara secuencias de palabras. Bajo estas circunstancias, se obliga al reconocedor del habla a tomar más frases situadas antes y después en el tiempo de manera iterativa, hasta que se consigue la resincronización.

Para obtener los resultados, no se tienen en cuenta estas frases correctamente detectadas que se fuerzan incorrectas para garantizar la sincronización, sino que se omiten. Solamente se trabaja con las secuencias incorrectas.

Para cada tipo de programas se muestran dos histogramas. En el primero de ellos, el eje horizontal hace referencia al número de segmentos erróneos por programa y el eje vertical al número de programas totales que hay de esa serie de programas. El segundo, se centra en las palabras por segmento. El eje horizontal y el eje vertical hacen referencia al número de palabras erróneas por segmento y el número de segmentos estudiados, respectivamente.

Los resultados de cada uno de los programas son los siguientes:

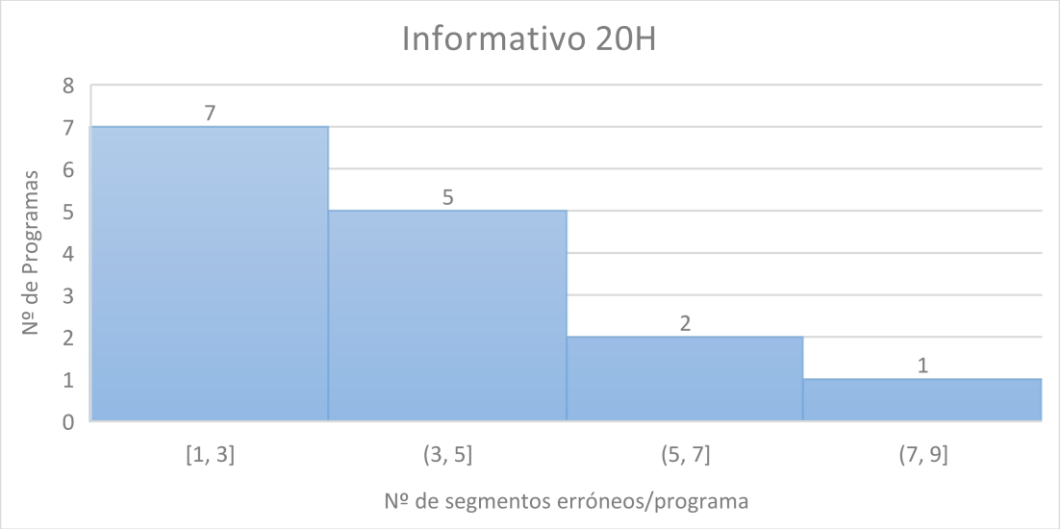

Figura 11. Nº de segmentos erróneos por programa. Informativo 20H

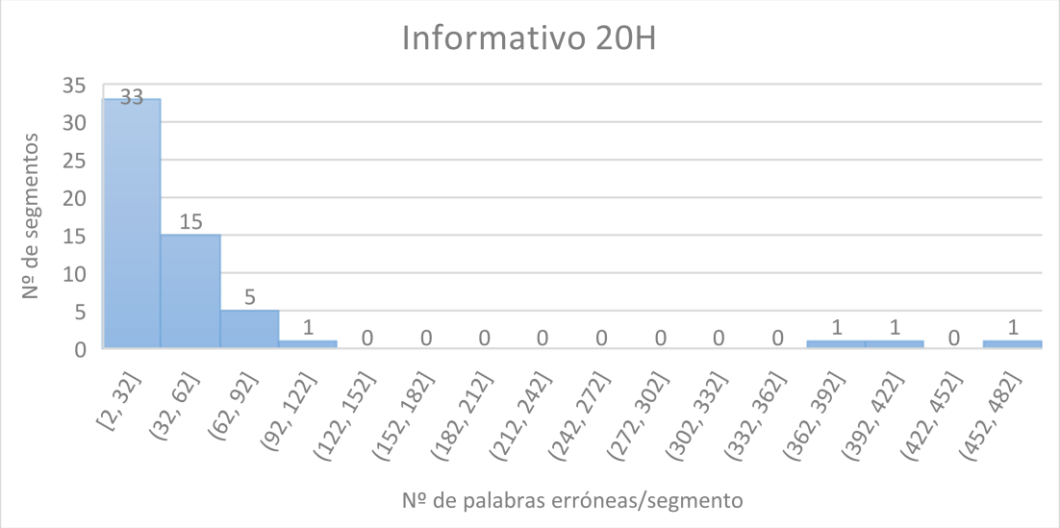

Figura 12. Nº de palabras erróneas por segmento. Informativo 20H

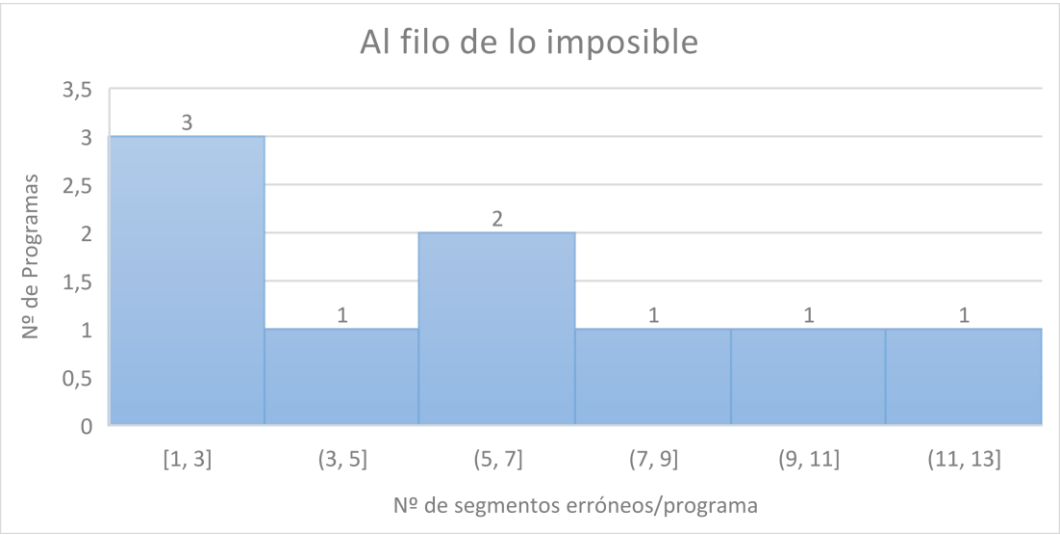

Figura 13. Nº de segmentos erróneos por programa. Al filo de lo imposible

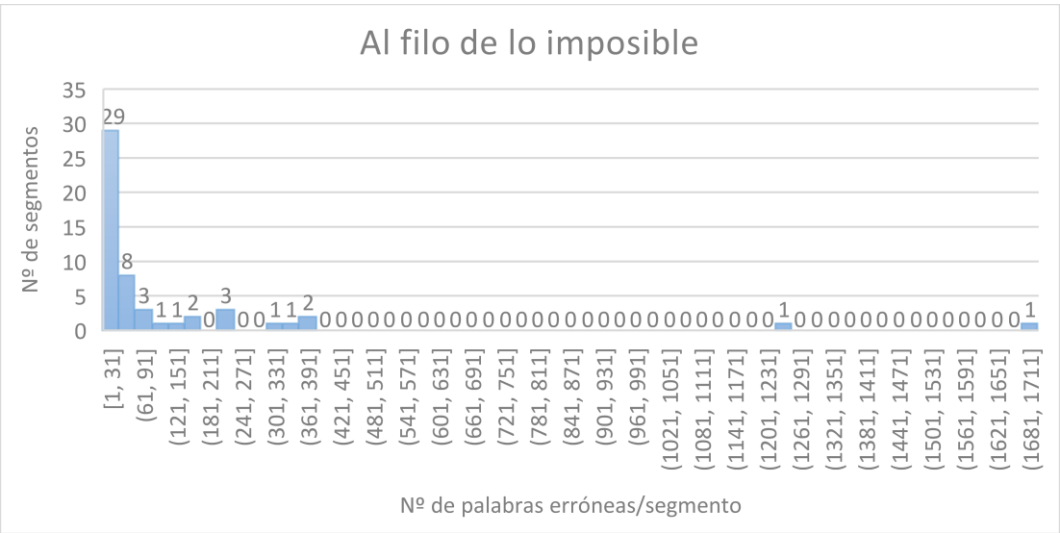

Figura 14. Nº de palabras erróneas por segmento. Al filo de lo imposible

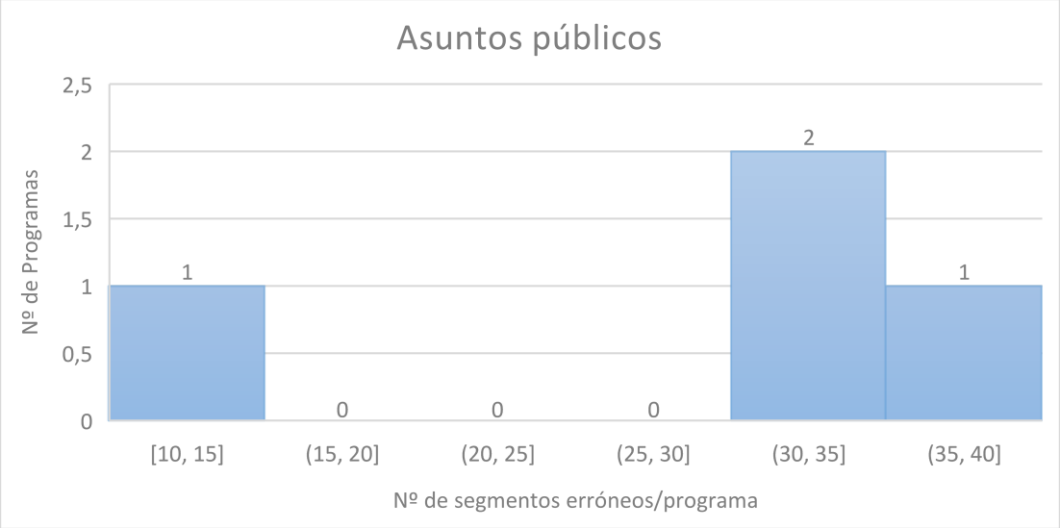

Figura 15. Nº de segmentos erróneos por programa. Asuntos públicos

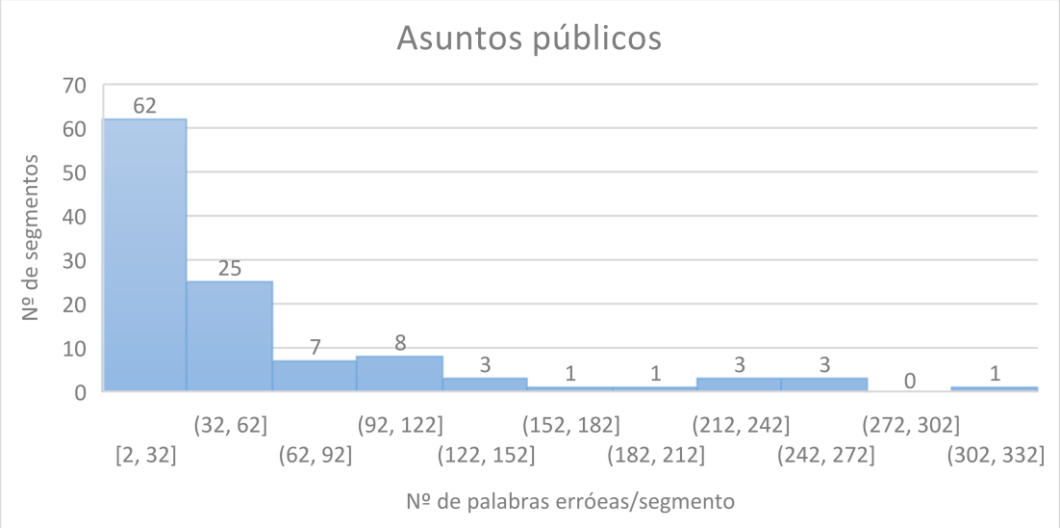

Figura 16. Nº de palabras erróneas por segmento. Asuntos públicos

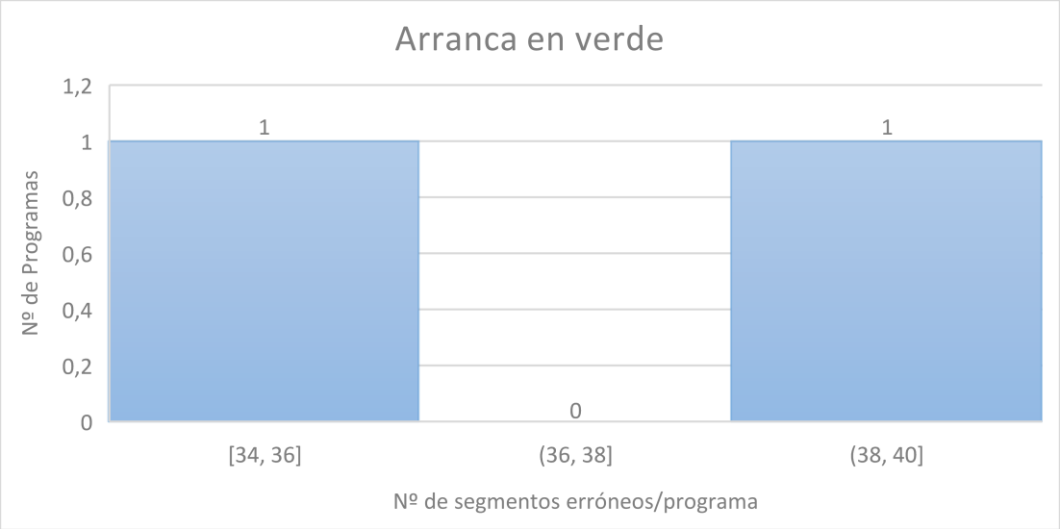

Figura 17. Nº de segmentos erróneos por programa. Arranca en verde

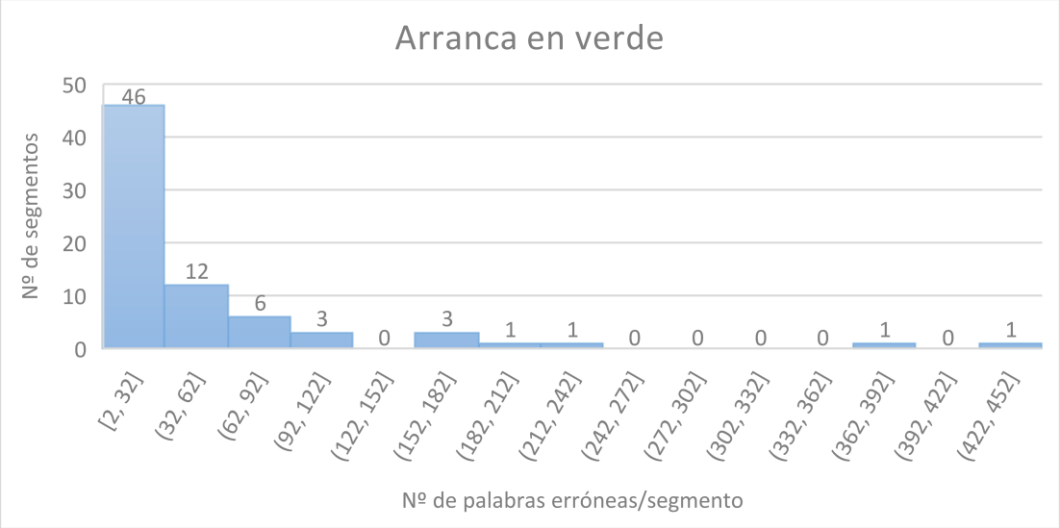

Figura 18. Nº de palabras erróneas por segmento. Arranca en verde

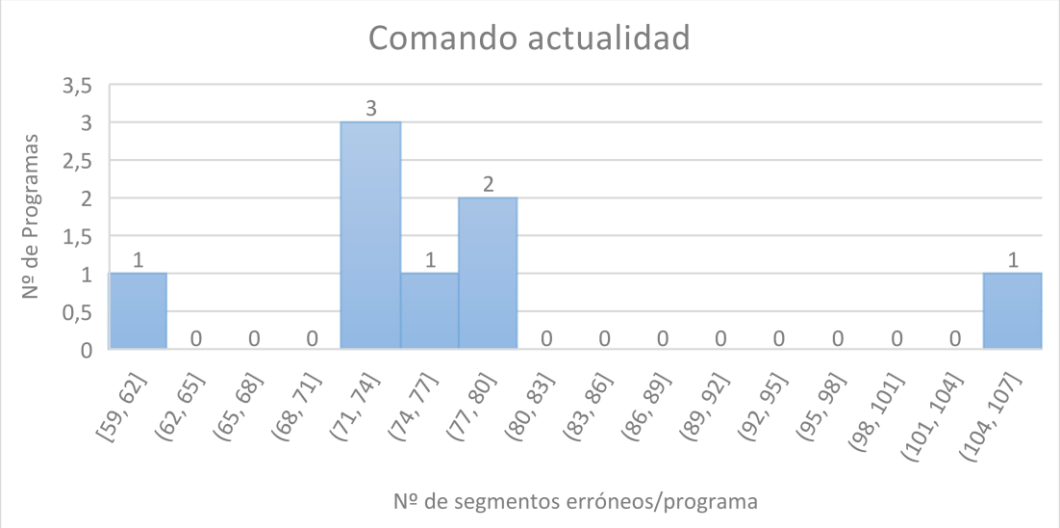

Figura 19. Nº de segmentos erróneos por programa. Comando actualidad

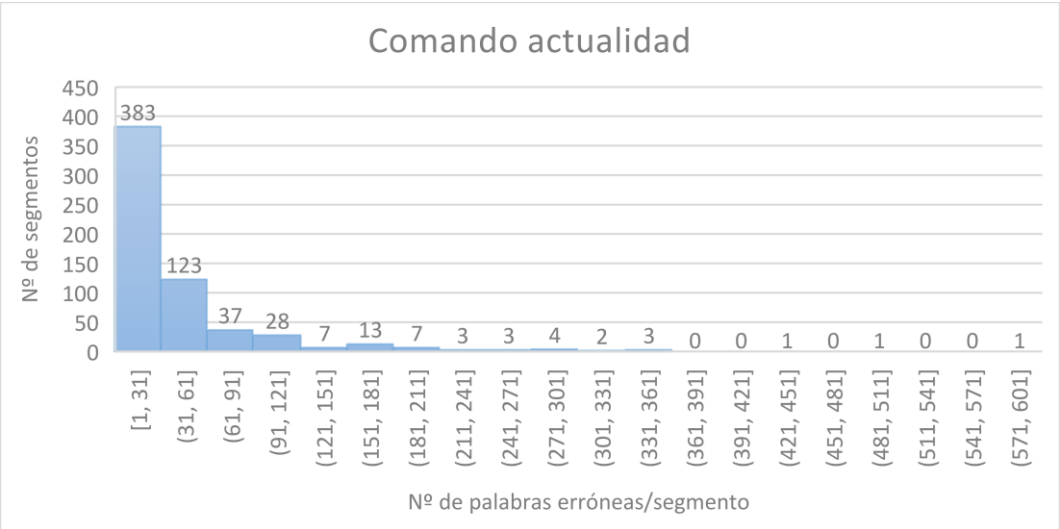

Figura 20. Nº de palabras erróneas por segmento. Comando actualidad

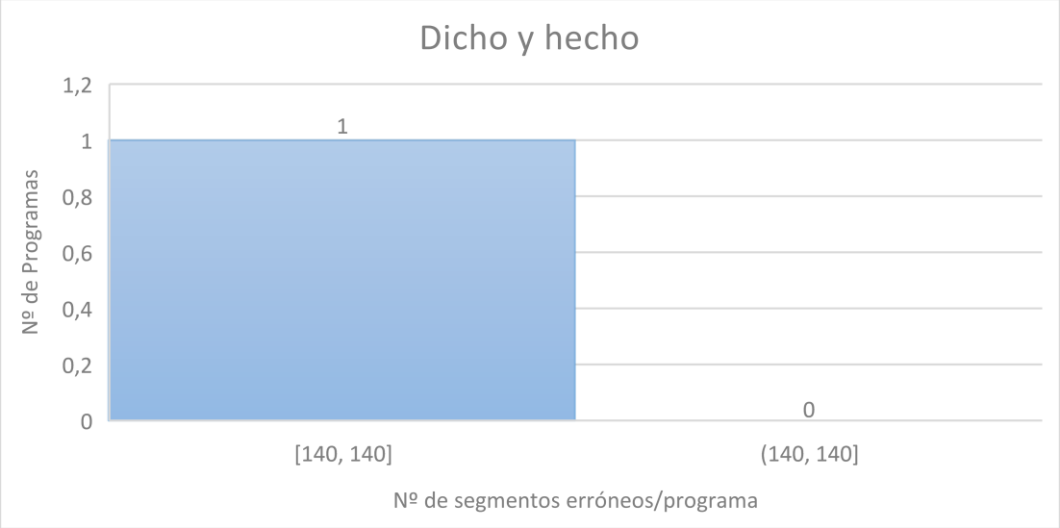

Figura 21. Nº de segmentos erróneos por programa. Dicho y hecho

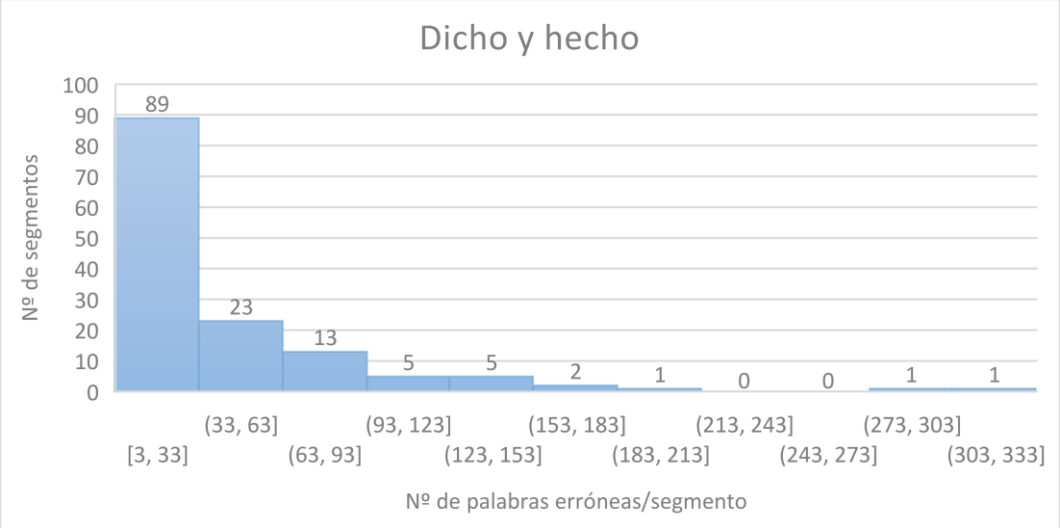

Figura 22. Nº de palabras erróneas por segmento. Dicho y hecho

Figura 23. Nº de segmentos erróneos por programa. España en comunidad

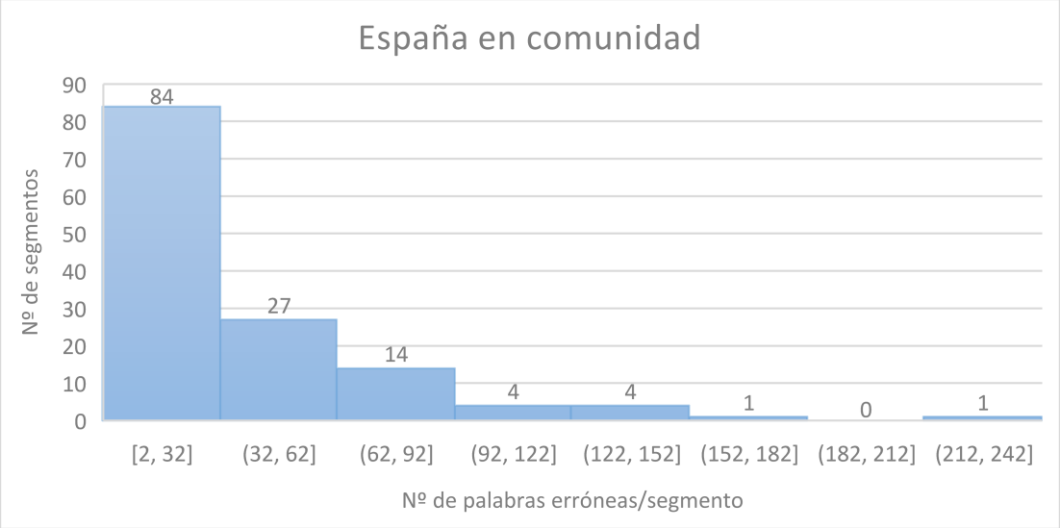

Figura 24. Nº de palabras erróneas por segmento. España en comunidad

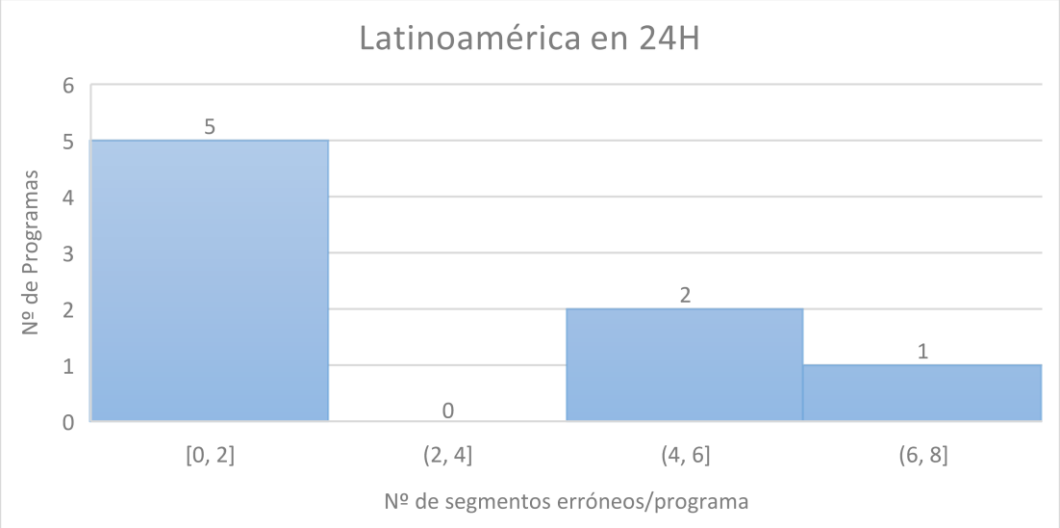

Figura 25. Nº de segmentos erróneos por programa. Latinoamérica en 24H

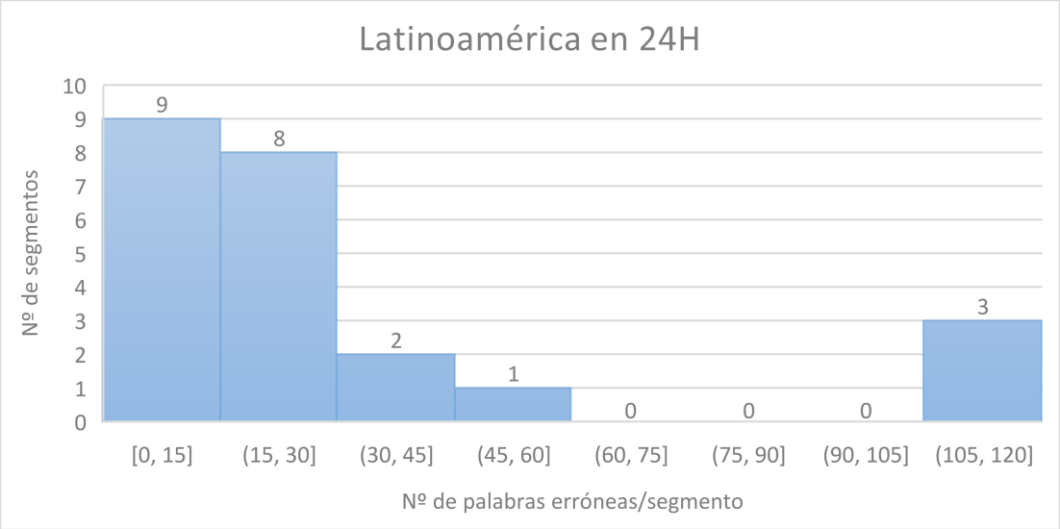

Figura 26. Nº de palabras erróneas por segmento. Latinoamérica en 24H

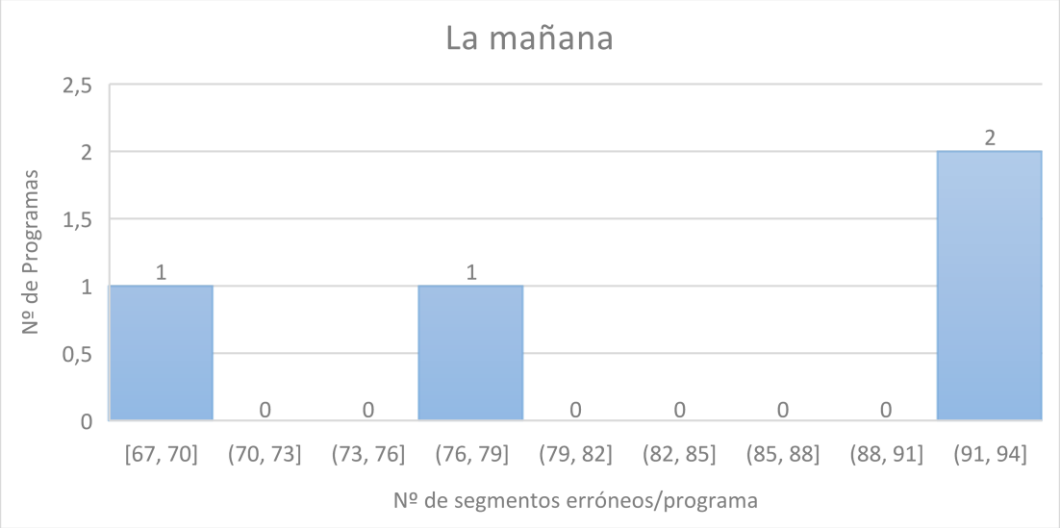

Figura 27. Nº de segmentos erróneos por programa. La mañana

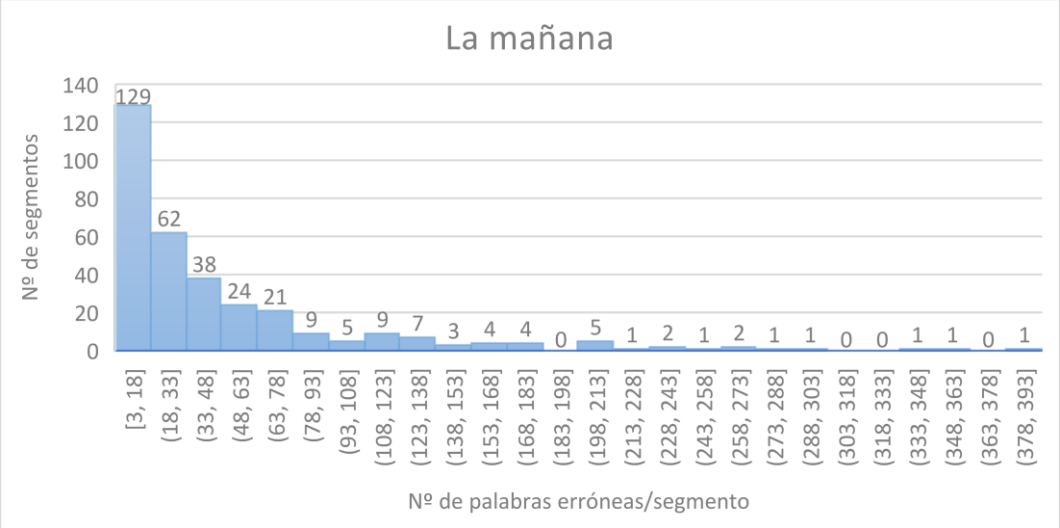

Figura 28. Nº de palabras erróneas por segmento. La mañana

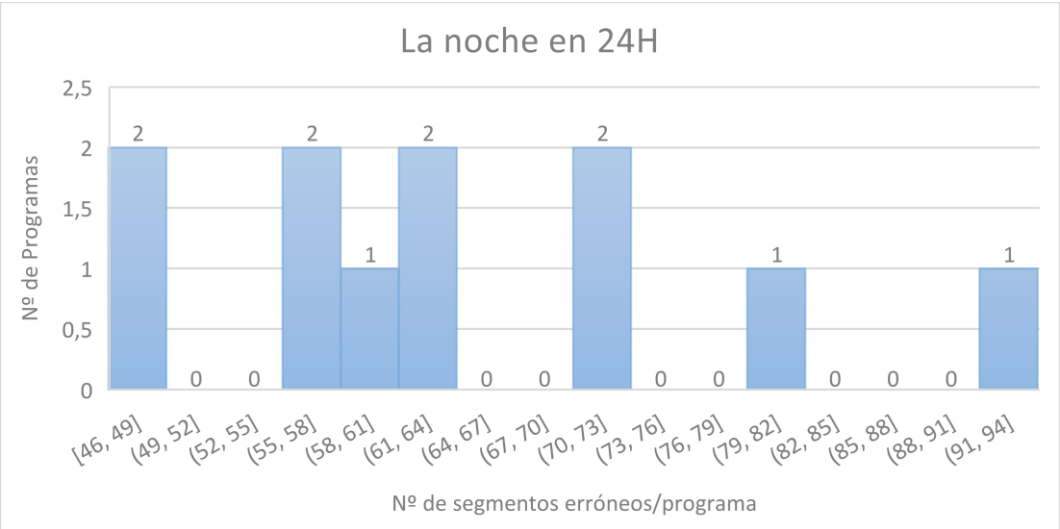

Figura 29. Nº de segmentos erróneos por programa. La noche en 24H

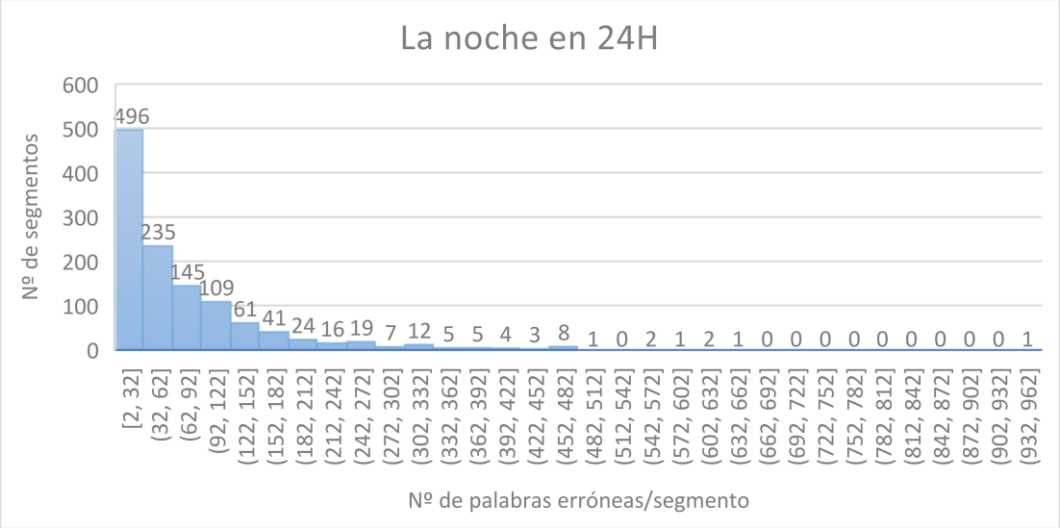

Figura 30. Nº de palabras erróneas por segmento. La noche en 24H

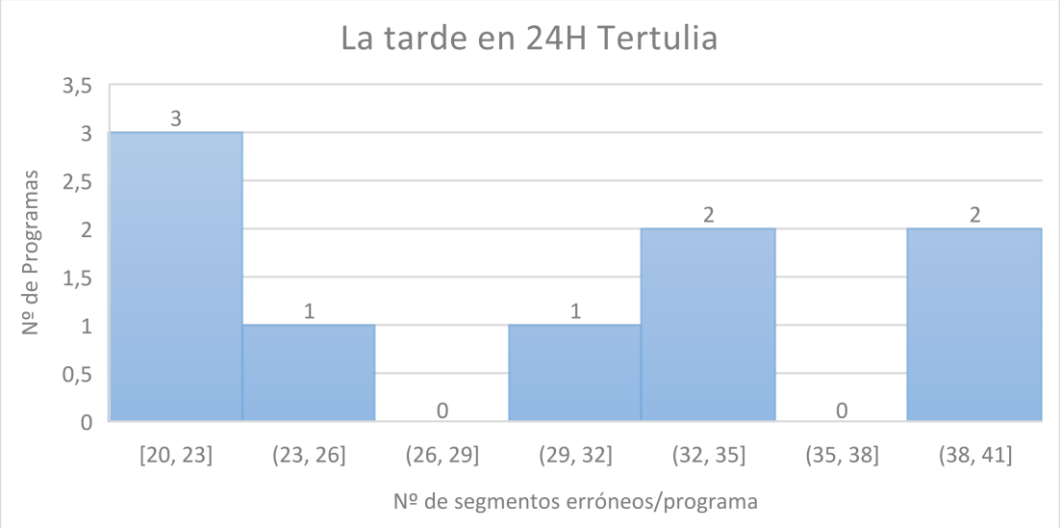

Figura 31. Nº de segmentos erróneos por programa. La tarde en 24H Tertulia

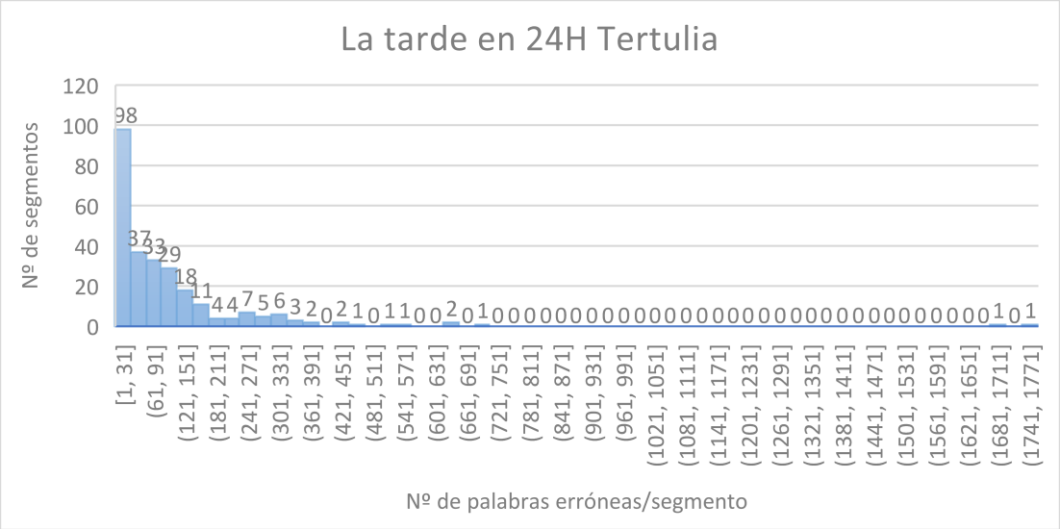

Figura 32. Nº de palabras erróneas por segmento. La tarde en 24H Tertulia

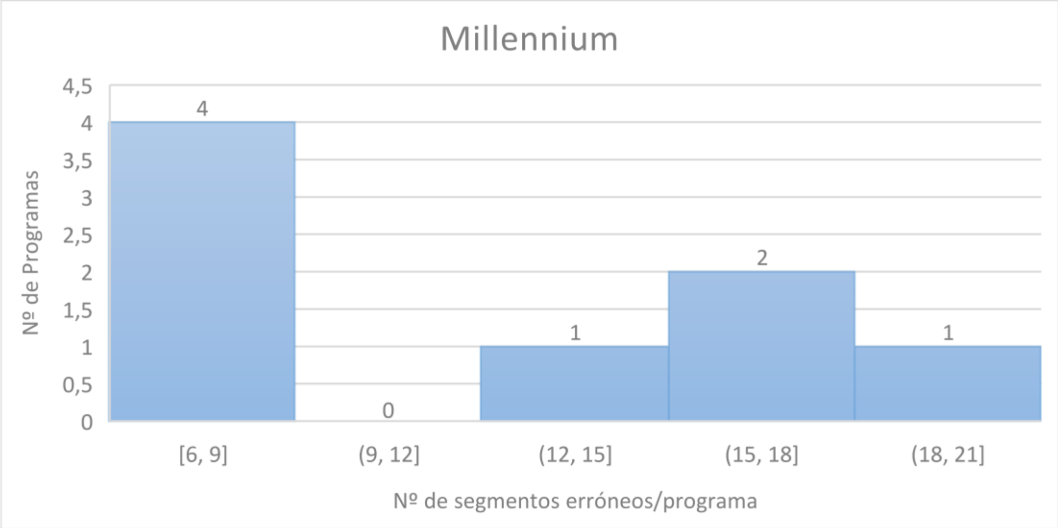

Figura 33. Nº de segmentos erróneos por programa. Millennium

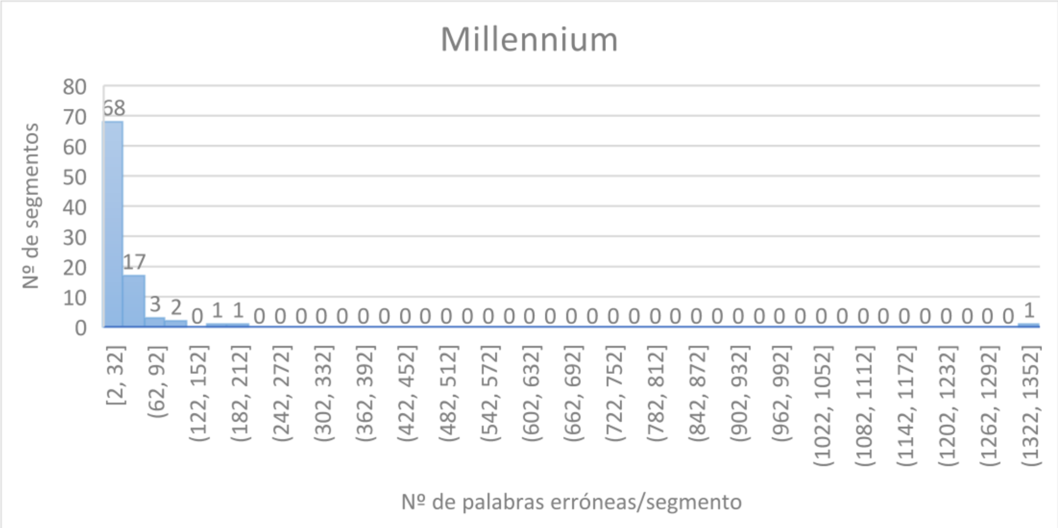

Figura 34. Nº de palabras erróneas por segmento. Millennium

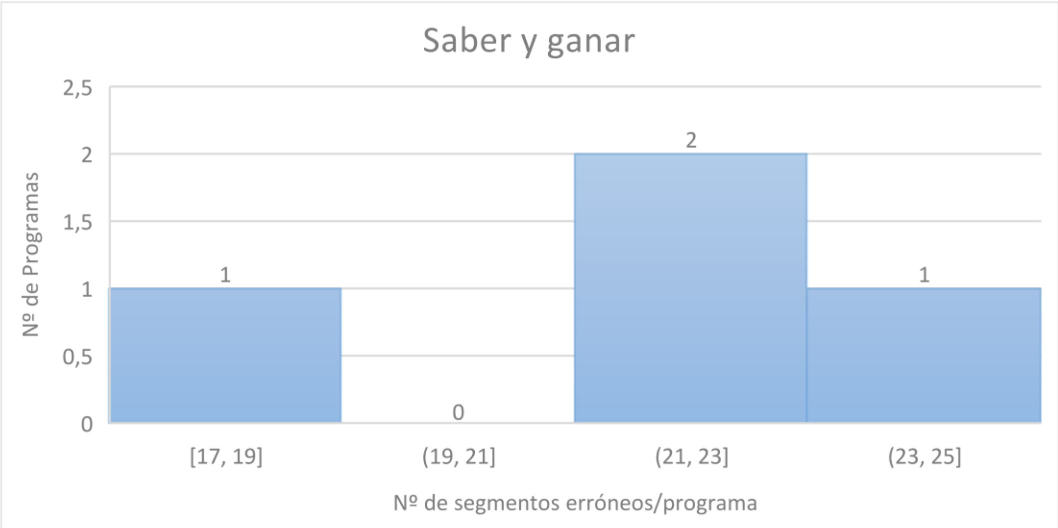

Figura 35. Nº de segmentos erróneos por programa. Saber y ganar

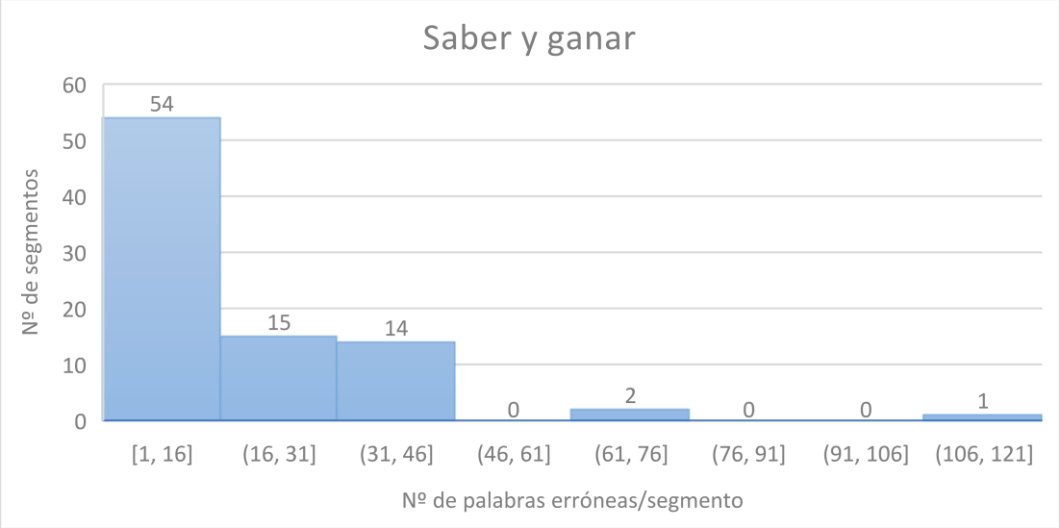

Figura 36. Nº de palabras erróneas por segmento. Saber y ganar

Comentando el primer tipo de gráfica, el histograma donde se refleja el número de segmentos erróneos por programa, se puede ver que, en general, se sigue la regla entre la calidad acústica y la dificultad, es decir, los ficheros con un audio más limpio cuentan con un menor número de segmentos erróneos. De hecho, "Latinoamérica en 24H" produce pocos fragmentos erróneos, destacando un programa concreto donde no se han producido errores. Otros programas como "Informativo 20H" o "España en comunidad" también producen pocos segmentos malos. Por el contrario, "Dicho y hecho", "Comando actualidad" o "La mañana" conllevan un número considerable de segmentos erróneos. La cantidad de estos segmentos, aunque sí está relacionada con la duración del audiovisual, está mayormente relacionada con la calidad acústica. Este hecho se demuestra comparando los segmentos erróneos de "Millennium" contra los segmentos de "Comando actualidad", donde a pesar de tener la misma duración aproximada, el número de zonas erróneas por programa es muy diferente. Sin embargo, la cantidad de segmentos errados no es especialmente importante a la hora de implementar el sistema, ya que se sigue el mismo tratamiento independientemente del número de zonas conflictivas.

Una vez que se conocen estos segmentos de texto y audio, se pueden obtener su longitud en palabras y ofrecer un estudio más detallado. No se tienen en cuenta los puntos y comas, solamente las palabras, para intentar comparar entre segmentos de una manera más exacta. Realizando una reflexión sobre el segundo tipo de histograma, en todos los tipos de programas se puede asegurar que la mayoría de los fragmentos rondan las 30 palabras, es decir, los fragmentos no reconocidos en la primera fase son cortos. Conforme nos alejamos de las 30 primeras palabras, el número de segmentos con longitud mayores de palabras va disminuyendo.

En un porcentaje bajo de casos, las longitudes de los segmentos son considerablemente mayor. Estos extractos pueden deberse a una zona especialmente difícil de sincronizar desde un principio o a las zonas en las que se necesita incluir frases anteriores y posteriores al segmento erróneo porque, aunque las frases estaban escritas en el sitio correcto, no estaban adecuadamente situadas en el tiempo.

Hay que comentar un caso especial, un programa de "La mañana" donde se deben omitir varios subtítulos en el fichero de subtítulos final debido a que un fragmento de texto no pudo ser sincronizado por el reconocedor, ya que este trabaja de forma secuencial y ese extracto concreto de

audio contaba con un solape entre hablantes prolongado, por lo que el reconocedor no era capaz de dar un fichero de salida.

CAPÍTULO 6. Conclusiones

Como se ha podido ver, se han tratado con diferentes tipos de programas de RTVE, lo que implica distintas calidades acústicas. Este hecho ha supuesto que la dificultad de tratamiento de los archivos audiovisuales es diferente según el tipo de programa. Esto no trae consecuencias negativas, sino que implícitamente conlleva una necesidad de robustez del sistema para intentar soportar todo tipo de programas.

Como resultado de la experimentación, se ha concluido que la primera fase de reconocimiento da mejores resultados con el uso de una gramática basada en estados finitos, que a su vez debe ser creada por la transcripción individual del archivo audiovisual a tratar. Hay que remarcar que este sistema solo es útil para los casos donde se dispone de esas transcripciones. En la segunda fase de reconocimiento donde se deben manipular los segmentos erróneos y volver a sincronizarlos, se consigue rehacer esa parte del texto, sin pérdidas, salvo casos excepcionales debidos a solapes prolongados entre hablantes como se ha mencionado anteriormente.

La motivación de este TFG ha sido la mejora del trabajo previo, tratar de sincronizar un archivo audiovisual con las transcripciones correspondientes mediante gramática forzada exclusivamente. Una vez finalizada la experimentación y obtener los resultados, sí se ha conseguido el objetivo de perfeccionamiento del trabajo anterior. Puede ocurrir que, en zonas especialmente conflictivas donde haya un ruido considerable u otras perturbaciones, los subtítulos no se coloquen adecuadamente en el tiempo, pero lo que sí se puede decir de una manera fiable es que, entre zonas conflictivas, sí habrá frases correctamente colocadas y, por lo tanto, se evita que estas zonas erróneas desfasen el sistema de subtítulos completo. De este modo, permitir al reconocedor poderse reenganchar en estas situaciones. Como resumen se puede decir que no se produce un retraso global entre subtítulos gracias a esta primera fase con libertad de creación de frases y la posterior corrección entre puntos de anclaje, que garantiza el orden entre subtítulos.

Sin embargo, no se pueden medir con exactitud este progreso ya que no se cuenta con los ficheros de subtítulos con tiempos correctamente situados para poder realizar una comparación. Bajo estas circunstancias, la alternativa es visualizar los subtítulos y el audio o vídeo del programa simultáneamente.

CAPÍTULO 7. Bibliografía

[1] Xuedong Huang, Alex Acero y Hsiao-Wuen Hon, “Spoken Language Processing. A Guide to Theory, Algorithm, and System Development”, 2001.

[2] Dan Jurafsky y James H. Martin, “Speech and Language Processing” (3rd ed. Draft), 2019.

https://web.stanford.edu/~jurafsky/slp3/

[3] Richard Bellman, “The theory of dynamic programming”, 1952.

https://www.ams.org/journals/bull/1954-60-06/S0002-9904-1954-09848-8/S0002-9904-1954-09848-8.pdf

[4] “SRILM - The SRI Language Modeling Toolkit”, 2019. (Visitado el día 15-05-2020)

http://www.speech.sri.com/projects/srilm/

[5] “Cátedra RTVE de la Universidad de Zaragoza” (Visitado el día 15-06-2020)

http://catedrartve.unizar.es/

[6] Jonathan Hui, “Speech Recognition — Feature Extraction MFCC & PLP”, 2019. (Visitado el día 11-06-2020)

https://medium.com/@jonathan_hui/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9

Índice de figuras

Índice de tablas

ANEXOS

ANEXO A

Modelos ocultos de Markov

Cadena de Markov

Los modelos ocultos de Markov [1] se basan en el concepto de cadena de Markov. La cadena de Markov está compuesta por variables aleatorias X, obtenidas de un alfabeto dado. Este alfabeto, van a ser palabras, aunque podrían ser otro tipo de variables. Una secuencia está compuesta por una sucesión de variables. La probabilidad de obtener una secuencia en concreto se muestra en la ecuación A.1.

$$P(x_1, x_2, x_3, \dots x_n) = P(x_1) \prod_{i=2}^{n} P(x_i \mid x_1, x_2, \dots x_{i-1}) \quad \text{(A.1)}$$

En este método se supone que $P(x_i \mid x_1, x_2, \dots x_{i-1}) = P(x_i \mid x_{i-1})$. Por tanto, la cadena de Markov de primer orden será:

$$P(x_1, x_2, x_3, \dots x_n) = P(x_1) \prod_{i=2}^{n} P(x_i \mid x_{i-1}) \quad \text{(A.2)}$$

Estas variables se pueden asociar a estados para un instante de tiempo dado. Es decir, que el estado actual en un momento concreto solamente depende del estado anterior.

$$P(x_i \mid x_{i-1}) = P(q_t = m \mid q_{t-1} = \mathrm{n}) \quad \text{(A.3)}$$

A continuación, se muestra dos parámetros más de la cadena de Markov:

$$a_{nm} = P(q_t = m \mid q_{t-1} = \mathrm{n}) \quad \text{(A.4)}$$

$$\pi_n = P(q_1 = m) \quad \text{(A.5)}$$

Donde a_{nm} es la probabilidad de pasar del estado n al estado m y π_n es la probabilidad de que la cadena de Markov comience en el estado n. Tanto n como m pueden tomar como valor todos los estados posibles.

Se deben cumplir las siguientes condiciones, conociendo el número de estados, N:

- Dado un m, $\sum_{m=1}^{N} a_{nm} = 1$ (A.6)
- $\sum_{n=1}^{N} \pi_n = 1$ (A.7)

Modelos ocultos de Markov

Los modelos ocultos de Markov (HMM) [2] son útiles cuando no tenemos una secuencia de eventos observables, como era el caso de la cadena de Markov. En este caso se busca determinar los eventos ocultos a partir de los que sí son observables. Los estados en sí permanecen ocultos pero no su salida, la cual se obtiene con una función probabilística asociada al estado. El reconocedor toma como función probabilística una combinación lineal de gaussianas y las observaciones son los vectores MFCC resultantes del bloque de procesamiento de señal.

Los componentes de modelo oculto de Markov son los siguientes:

- Q = $q_1, q_2, \dots q_N$ → Conjunto de N estados.
- A = $a_{11}, \dots a_{nm}, \dots a_{NN}$ → Probabilidad de transición de un estado n al estado m. Remarcar que en el modelo oculto de Markov también debe cumplirse que $\sum_{m=1}^{N} a_{nm} = 1$.

- O = o_1, o_2, …o_T → Grupo de T observaciones acústicas (vectores MFCC).
- B = $b_n(o_t)$ → Probabilidades de observación. Probabilidad de generar una determinada observación o_t por un estado n.
- π = π_1, π_n, … π_N → Probabilidad de que el modelo comience en el estado n. Al igual que la cadena de Markov, debe cumplirse que $\sum_{n=1}^{N} \pi_n = 1$.

El modelo de primer orden, a la par que la cadena de Markov, se debe asegurar la **condición de Markov** tal que

$$P(q_i|\, q_1, q_2, \ldots q_{i-1}) = P(q_i|\, q_{i-1}) \quad \text{(A.8)}$$

También hay **independencia de salida**, la observación de salida o_i solamente depende del estado que la produce, q_i:

$$P(o_i|q_1, \ldots q_i, \ldots q_T, o_1, \ldots o_i, \ldots o_T) = P(o_i|\, q_i) \quad \text{(A.9)}$$

ANEXO B

Fichero de referencias de tiempo (extensión .stm)

El formato STM (Segment Time Marked) [5] es una concatenación de registros de segmentos de texto de un archivo de audio. Los registros están separados con saltos de línea. Este formato contiene:

- Nombre del archivo audiovisual al que hace referencia.
- Identificador del canal.
- Identificación del hablante.
- Intervalo de tiempo en que se reproduce el registro.
- Posibles etiquetas adicionales.
- El texto en sí.

```
20H-20180202 1 #_0 2676.520020 2678.500000 <,,> Gracias y muy buenas noches .
20H-20180202 1 #_0 2678.500000 2679.199951 <,,> Hasta el lunes .
```

Figura 37. Fichero STM (Segment Time Marked)

Fichero de subtítulos (extensión .srt)

El formato Subrip [5] es un fichero de texto con extensión .srt, el formato con el que se muestran los resultados finales obtenidos en este TFG. Se escoge este formato porque es un archivo de subtítulos ampliamente extendido.

El fichero en sí consta de:

- Número que identifica a cada subtítulo.
- Intervalo de tiempo en el que se muestran el subtítulo por pantalla.
- El subtítulo.
- Línea en blanco que indica el comienzo del siguiente subtítulo.

```
1
00:00:19,139 --> 00:00:20,950
pero tenemos que aprender
a hacerlo mejor.

2
00:00:21,430 --> 00:00:23,20
Conductores, peatones, ciclistas.

3
00:00:23,129 --> 00:00:25,40
Todos tenemos que aprender
a convivir y, para eso,

4
00:00:25,49 --> 00:00:26,930
hay que usar
una herramienta maravillosa
```

Figura 38. Formato Subrip de subtítulos

ANEXO C

DTW (dynamic time warping)

Dynamic Time Warping [1] se emplea para medir la distorsión entre dos secuencias dadas. Esta distorsión se mide a partir de dos vectores característicos alineados de manera que el coste total de la distancia entre las secuencias proporcionadas sea mínimo. Estos vectores se pueden modificar tal y como se puede ver en la figura 9, para cumplir la distancia mínima entre secuencias. Asociando cada par (i, j) a una distancia d(i, j) siendo x_i e y_j los vectores característicos, y teniendo en cuenta que x e y tienen longitud N y M respectivamente, se puede hallar la distancia acumulada de todas las posibles combinaciones de los vectores x e y, comenzando por (1, 1) hasta (N, M). Todas estas rutas serán exponenciales y entre todas las posibles rutas obtenidas hasta llegar a D(N, M), se toma la óptima, la menor. Para reducir el coste de computación, se eliminan las rutas que no son óptimas, basándose en el principio de programación dinámica. Dado que la ruta óptima en cada paso depende del paso anterior, la distancia mínima en cada paso se obtiene cumpliendo la siguiente ecuación:

$$D(i, j) = \min_{k} \left[D(i-1, k) + d(k, j) \right] \quad \text{(C.1)}$$

Se puede observar el proceso de una manera más visual en la figura 39.

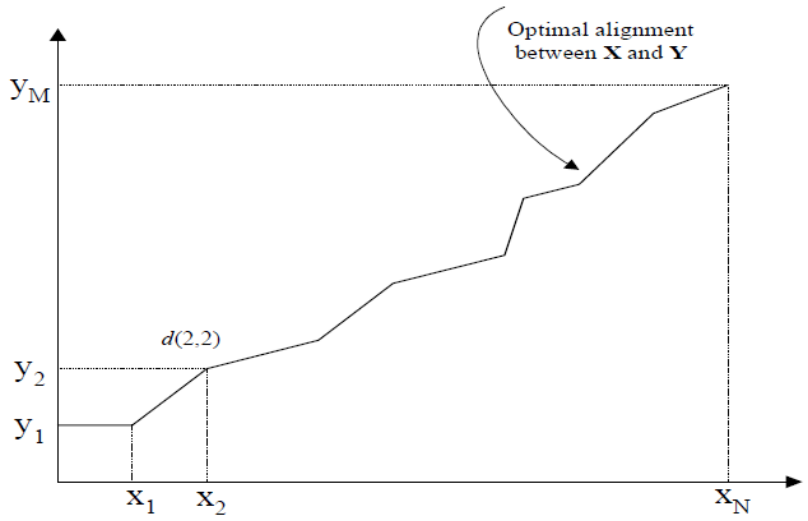

Figura 39. DTW. Comparación entre la secuencia de palabras x e y [1]

Como se muestra en la figura 39, solo es necesario considerar el menor coste para cada par (i, j) dados M posibles movimientos. Cada vez que encontremos un par óptimo (i, j) se pueden guardar los índices en una tabla de punteros B(i, j), para rastrear la ruta una vez hemos obtenido D(N, M). El algoritmo DTW se muestra en la figura 40.

Step 1: Initialization

$D(1,1) = d(1,1), B(1,1) = 1$, for $j = 2,\ldots,M$ compute $D(1,j) = \infty$

Step 2: Iteration

for $i = 2,\ldots,N$ {

for $j = 1,\ldots,M$ compute {

$$D(i,j) = \min_{1 \le p \le M} \left[D(i-1,p) + d(p,j) \right]$$

$$B(i,j) = \arg\min_{1 \le p \le M} \left[D(i-1,p) + d(p,j) \right] \quad \}\}$$

Step 3: Backtracking and Termination

The optimal (minimum) distance is $D(N,M)$ and the optimal path is $(s_1, s_2, \ldots, s_N)$

where $s_N = M$ and $s_i = B(i+1, s_{i+1})$, $i = N-1, N-2, \ldots, 1$

Figura 40. Algoritmo DTW [1]

También es importante destacar que inicializar D(1, j)=∞ es significativo, para que el algoritmo no continúe por esa ruta, ya que estaríamos comparando la secuencia con ella misma y no frente a la otra secuencia facilitada.